2004-09

NPSAT1 magnetic attitude control system algorithm verification, validation, and air-bearing tests

Herbert, Eric W.

Monterey California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM ALGORITHM VERIFICATION, VALIDATION, AND AIR–BEARING TESTS

by

Eric W. Herbert
September 2004

Co–Thesis Advisors: Barry Leonard
Xiaoping Yun

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: NPSAT1 Magnetic Attitude Control System Algorithm Verification, Validation, and Air Bearing Tests			**5. FUNDING NUMBERS** NRO RSP-38
6. AUTHOR(S) Herbert, Eric, W.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8.PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Reconnaissance Office Naval Postgraduate School Space Systems Academic Group			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

NPSAT1 is a gravity–gradient friendly, prolate body designed to fly at 600 ±40 km inclined to 34.5 degrees. The satellite uses a magnetic 3-axis active attitude control system (ACS) using magnetic torque rods that interact with the Earth's magnetic field. This thesis accomplishes three goals.

The first objective was to verify and to validate the magnetic attitude control system program and model developed by Leonard. The verification and validation process was completed in two steps. The first step accomplished an independent modeling of the Earth's magnetic field using MATLAB. The second step completed a verification via inspection of Leonard's ACS SIMULINK model. The verification confirmed that Leonard's modular sub–components of the disturbance torques, the quaternion vectors, the Euler angles, the spacecraft kinematics and dynamics, and the ACS control laws conformed to current ACS empirical theory.

The second goal was to establish a laboratory used to demonstrate the ACS robustness and ability to perform as designed. The laboratory was created to house an air-bearing platform that simulates NPSAT1 characteristics.

The third goal was to perform *hardware-in-the-loop* experiments with the NPSAT1 ACS software and model. *Hardware-in-the-loop* tests were performed to the magnetic torque rods, torque rod driver circuit board, micro-controller computer, and control interfaces. Specifically, solenoid current tests, magnetic field determination tests, and digital-to-analog conversion tests were completed.

14. SUBJECT TERMS Magnetic Attitude Control, Torque Rods, 3-Axis Active Control, Magnetometers, Earth Magnetic Field, Steering Control Law, Bdot Control Law, Quaternion, Small Satellites, Low Earth Orbit Satellites, Hardware-in-the-Loop			**15. NUMBER OF PAGES 211**
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

THIS PAGE INTENTIONALLY LEFT BLANK

NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM ALGORITHM VERIFICATION, VALIDATION, AND AIR–BEARING TESTS

Eric W. Herbert
Lieutenant Commander, United States Navy
B.A., Pennsylvania State University, 1991
M.S., Naval Postgraduate School, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2004

Author: Eric W. Herbert

Approved by: Professor Barry S. Leonard
Co-Advisor

Professor Xiaoping Yun
Co-Advisor

John P. Powers
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

NPSAT1 is a gravity–gradient friendly, prolate body designed to fly at 600 ±40 km inclined to 34.5 degrees. The satellite uses a magnetic 3-axis active attitude control system (ACS) using magnetic torque rods that interact with the Earth's magnetic field. This thesis accomplishes three goals.

The first objective was to verify and to validate the magnetic attitude control system program and model developed by Leonard. The verification and validation process was completed in two steps. The first step accomplished an independent modeling of the Earth's magnetic field using MATLAB. The second step completed a verification via inspection of Leonard's ACS SIMULINK model. The verification confirmed that Leonard's modular sub–components of the disturbance torques, the quaternion vectors, the Euler angles, the spacecraft kinematics and dynamics, and ACS control laws conformed to current ACS empirical theory.

The second goal was to establish a laboratory used to demonstrate the ACS robustness and ability to perform as designed. The laboratory was created to house an air-bearing platform that simulates NPSAT1 characteristics.

The third goal was to perform *hardware-in-the-loop* experiments with the NPSAT1 ACS software and model. *Hardware-in-the-loop* tests were performed to the magnetic torque rods, torque rod driver circuit board, micro-controller computer, and control interfaces. Specifically, solenoid current tests, magnetic field determination tests, and digital-to-analog conversion tests were completed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS AND ACRONYMS

ACS	attitude control subsystem
ATJ	Advanced Triple Junction
b	normalized magnetic field in orbit frame
B	normalized magnetic field in body frame
Bdot	derivative of *B* with respect to time
B_r, B_ϕ, B_θ	body frame components of *B* in spherical coordinates
B_x, B_y, B_z	body frame components of *B* in rectangular coordinates
B^2	*B* magnitude
C&DH	command and data handling subsystem
CERTO	coherent electromagnetic radio tomography
cm	center of mass
cp	center of pressure
CPE	configurable processor experiment
COTS	commercial-off-the-shelf
DCM	direction cosine matrix
dL	cm / cp offset
DOD	Department of Defense
DPS	digital process server
EELV	expendable launch vehicle
EPS	electrical power subsystem
ESPA	EELV secondary payload adapter
g_x, g_y, g_z	actuator time average gains
I_x, I_y, I_z	principal moments of inertia
IAGA	International Association of Geomagnetism and Aeronomy
IGRF	International Geomagnetic Reference Field
I/O	input / output
k	Bdot gain
K	controller gain, $[K_a\ K_b]$
K_m	field "dipole strength" ~ Tesla
L_r	reduced order estimator gain (3x3)
LEO	low earth orbit
m_p	magnetic moment produced
m_r	magnetic moment requested
MEMS	micro-electromechanical system
MIMO	multiple inputs-multiple outputs
MOI	moments of inertia
NGDC	National Geophysical Data Center
NOAA	National Oceanic and Atmospheric Administration
NPS	Naval Postgraduate School
NRL	Naval Research Laboratory
PANSAT	Petite Amateur Satellite
q1, q2, q3, q4	quaternion elements
RAM	random access memory

RFS	radio frequency subsystem
ROE	reduced order estimator
SISO	single input-single output
SMS	solar cell measurement system
SSAG	Space Systems Academic Group
STP	Space Test Program
T_p	torque produced by the torque rods
$\vec{x}$	state vector $\left(\varphi,\theta,\psi,\dot{\varphi},\dot{\theta},\dot{\psi}\right)$
$\vec{y}$	measurement vector $\left(\varphi,\theta,\psi\right)$
T_r	control law requested torque
TEC	total-electron-content
$\vec{u}$	control vector
VISIM	visible wavelength imager
α	true anomaly
β	angle between sun and orbit plane
Γ	density variation factor
ν	orbit angle WRT the sub-solar point
$\rho,\rho_o,\rho_{\max}$	atmospheric density, average, max
ϕ,θ,ψ	yaw, pitch, roll Euler angel sequence
ω	angular velocity WRT orbit frame
ω_e	Earth spin rate
ω_n	line of nodes regression rate
ω_o	orbit angular velocity
${}^{O}\omega^{B}$	body-to-orbit angular velocity
${}^{N}\omega^{B}$	body-to-inertial angular velocity
${}^{N}\omega^{O}$	orbit-to-inertial angular velocity

ACKNOWLEDGMENTS

First and foremost, gratitude must be expressed to the United States Navy. The United States Navy has placed a great deal of confidence in my abilities and potential by sending me to the Naval Postgraduate School, of which, I hope I can serve to their expectations.

Second, I express the deepest gratitude and appreciation to Professor Barry S. Leonard for the opportunity to conduct this research. The NPSAT1 research that I performed under his guidance allowed me to broaden my education by crossing the Space Systems Engineering and Electrical Engineering academic boundaries. Professor Leonard's tutelage, support, and advice allowed me to grow personally and professionally while in the pursuit of higher knowledge.

Third, acknowledgement of Dr. Xiaoping Yun's fantastic teaching style and steadfast support is a must. By asking the hard questions and providing timely advice, Dr.Yun helped me to develop critical problem solving skills and rational thought processes that were crucial in the development of this thesis.

Fourth, I wish to thank Professor Arthur Schoenstadt of the NPS Math Department and LCDR Robert Broadston, USN (RET) from the NPS Electrical Engineering Department. Both Professor Schoenstadt and LCDR Broadston provided outstanding assistance during MATLAB algorithm and SIMULINK model generation. Each spent countless hours assisting me in the review and debugging of code. Without their guiding hands, the bulk of the models generated in this thesis would not have been completed.

Lastly, I wish to thank the NPSAT1 Staff Engineers, Mr. Dan Sakoda, Mr. James Horning, Mr. Ronald Phelps, Mr. David Rigmaiden, and Mr. Glenn Harrell for the superb support during the thesis research process. Each played a most significant role in my being able to solve a technical or documentary issue.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

NPSAT1 is the second experimental small satellite designed and constructed by the Naval Postgraduate School. The satellite's mission is to afford officer–students an opportunity to apply classroom knowledge to a platform that will actually fly in space. The NPSAT1 project is administered by the NPS Space System's Academic Group and is sponsored by the DOD Space Test Program. The scheduled launch is October 2006.

NPSAT1 is a prolate body designed to fly at an altitude of 600 ± 40 km. NPSAT1 is a gravity–gradient friendly, non–spinning, nadir–pointed body inclined at 35.4 degrees. The satellite's attitude control system is unique and experimental. Specifically, NPSAT1 will use a three-axis active magnetic attitude control system designed to maintain the satellite within 10 degrees of its pointing target. The magnetic attitude control system is designed to change its attitude based upon measurement of the Earth's magnetic field. An attitude change is completed by comparing externally measured field values to approximated values resident on the spacecraft's computer. The attitude control system program uses this comparison to perform an estimated attitude error which equates to the amount of dipole moment needed to torque the body. The torque is affected by using three magnetic dipole moments that are aligned with the X, Y, and Z axes of the satellite. Design of the magnetic attitude control program and simulation model was first completed by Leonard and is located in Ref. [6].

This thesis completes three goals. The first goal was to accomplish an independent verification and validation of Leonard's attitude control system (ACS) program and model. The verification and validation process was completed in two steps. The first step accomplished an independent modeling of the Earth's magnetic field using MATLAB. The second step completed a verification via inspection of Leonard's ACS SIMULINK model. The verification confirmed that Leonard's modular sub–components of the disturbance torques, the quaternion vectors, the Euler angles, the spacecraft kinematics and dynamics, and ACS control laws conformed to current ACS empirical theory.

The second goal was to establish a laboratory that houses an air–bearing platform. The air–bearing is designed to mimic NPSAT1 attitude movement using the resident ex-

ternal magnetic field as a reference. The final objective was to commence *hardware-in-the-loop* tests that exercised the ACS program and supporting hardware.

The verification and validation via inspection of Leonard's ACS program and ACS model was completed satisfactorily. This was accomplished by researching the magnetic attitude control theory cited by Leonard and independently drafting an ACS program and model similar to Leonard's. Both Leonard's and this thesis' ACS programs included the modeling of the Earth's magnetic field, conversion of coordinates to the appropriate reference frame, calculation of the Euler 3-2-1 angular rotation scheme, identification and application of external disturbance torques, comparison of ephemeris field data to measured field strengths, determination of the required dipole moment, and application of counter torques. The program and model were generated using MATLAB and SIMULINK.

Following the successful verification of Leonard's program and model, a laboratory designed to test the ACS program was established. The laboratory was designed to use an air–bearing test platform.The air–bearing was constructed to exhibit the characteristics of NPSAT1. The air-bearing uses three magnetic torque rods capable of producing a dipole moment of $\pm 33\ \mathrm{A \cdot m^2}$. Additionally hardware located on the air-bearing includes a torque rod driver circuit board, a micro-controller single board computer, batteries, power supply, and remote communications equipment. However, since the ACS program uses external magnetic field values for comparison against measured values, a thorough understanding of the laboratory's resident magnetic field was required. Therefore, three sets of magnetic field surveys were conducted. These surveys provided the necessary data needed to map the laboratory's resident magnetic field. The mapping provides a description of the field's magnitude, its location and direction, and the presence of local field disturbances, if any. The information developed from the mapping process is used as the air-bearing ACS ephemeris data.

After the laboratory was established, *hardware-in-the-loop* tests commenced. The tests were designed to exercise the ACS program using a hardware environment. However, prior to executing the *hardware-in-the-loop* in the loop process, the ACS program had to be converted into an executable code format. This conversion process was

completed using the SIMULINK Real Time Workshop function. The Real Time Workshop function converts a model into ANSI 'C' code. This executable format code was then loaded to the air-bearing's micro-controller computer. Once loaded, tests commenced to determine if the ACS program could successfully apply a dipole moment.

The initial battery of experiments was marginally successful. The ACS program successfully read incoming magnetometer readings and created torque commands. However, repeated experiments proved that the output commands were unreliable. An examination of the executable code, signal routing, and ACS model variables was completed. No faults were recognized; therefore, attention was focused upon the air–bearing hardware. Specifically, current tests were completed to determine if the appropriate amount of current was being applied to the torque rods. The tests proved that the solenoids were not receiving the correct current values called for by the ACS. Inspections proved that the micro–controller was not routing the input/output data correctly and that there were faults residing in the torque rod driver board. Repairs were affected and testing resumed.

The initial *hardware-in-the-loop* tests showed that other problems that needed attention. Using bench support test equipment, the torque rod driver circuit board digital-to-analog process was measured. The results proved that the D/A conversion was being obscured by a great deal of noise. Thus, the analog values produced by the driver board were being skewed which caused the current to the solenoids to be misapplied. Furthermore, more research was performed upon the air–bearing's magnetic torque rods to gain a better understanding of the coil's capabilities. From this analysis, it was determined that the torque rods presently being used do not provide the fidelity needed to apply precise torque at the lower dipole moment values. Additionally, this thesis shows that torque rods presently in use maintain a residual magnetism that could possible upset the ACS process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THESIS OBJECTIVE

The Naval Postgraduate School (NPS) Space Systems Academic Group (SSAG) and the Electrical and Computer Engineering Department are currently building a Low Earth Orbit (LEO) satellite. The scheduled launch is October 2006. The satellite, NPSAT1, is being designed, constructed, and tested by NPS officer–students, faculty, and departmental engineering staff. The project's goal is to afford officer-students an opportunity to apply their engineering skills to a fully mission–capable space vehicle. The NPSAT1 mission and development was conceived and developed by the SSAG. The project is sponsored, executed, and funded by the Department of Defense (DOD Space Test Program (SMC Det 12)).

NPSAT1 is a prolate body that will fly at an altitude of 600 ± 40 km. It is a nadir–pointing, three–axis, non–spinning body inclined to an angle of 35.4 degrees. NPSAT1 uses a command and data handling (C&DH) subsystem, an electrical power (EPS) subsystem, an attitude control (ACS) subsystem, and a radio frequency (RFS) subsystem. Additionally, the satellite will house six experiments.

This thesis focuses upon the attitude control system. NPSAT1 utilizes a closed–loop magnetic actuating feedback control system using magnetometers and magnetic torque control rods. Most satellites rely upon a combination of momentum wheels, reaction wheels, thrusters, horizon sensors, sun sensors, or GPS to provide attitude control. However, with the availability of microcomputer processing, the use of magnetic torque rods and magnetometers to fully control a spacecraft's attitude is now feasible. NPSAT1 is unique. No other spacecraft has ever been flown using only magnetic torque rods to drive its attitude position.

The magnetic ACS relies upon a MATLAB program to measure and compare the Earth's magnetic field to an embedded database. If the magnetic field is not in line with a calculated value, a command will be sent to initiate attitude control correction using the magnetic torque rods.

To date, all research has been completed using a program and model [9] developed by Professor Barry Leonard of the NPS Space Systems Academic Group. However, a complete

independent verification via empirical calculations, computer simulations, and static methods had yet to be accomplished.

This thesis conducted that independent analysis, verification, and validation of the ACS control program and model. Additionally, this thesis documents the establishment of an air-bearing laboratory. The air–bearing laboratory was used to test the ACS control program using *hardware-in-the-loop* experiments. These goals were accomplished in four phases.

First, an independent derivation of the control program and model based upon the most current literature regarding attitude control systems using magnetic torques was performed. This analysis examined the resources Leonard cites to insure correctness and validity.

The program and model derivations used classical and state space control theory coupled with space system attitude control dynamics theory. The analysis included the application of the quaternion control law, reduced order estimators, gravity gradient, solar and aerodynamic disturbance torques. Additionally, the Earth's 10^{th}-order, 10^{th}-degree magnetic field was modeled for use during simulation and hardware tests.

Second, the work performed a detailed examination of Leonard's MATLAB/SIMULINK model was performed to verify that it is correct. The ACS model was confirmed to be accurate by inspecting Leonard's model for accuracy versus the research sources he cited.

Third, a laboratory that houses an air-bearing test platform was established. Specifically, the laboratory's local magnetic field was surveyed and mapped. The information gathered from the mapping process established ephemeris data required by the ACS program to compute a counter torque. Inducing current into three solenoid coils aligned with the spacecraft's three axes creates the counter torque. The current traveling through the solenoid's windings creates a dipole moment, magnetic field, and torque.

Last, an air–bearing platform was used to physically test the control model and program. Magnetometers and torque control rods were tested to ensure all software and hardware interfaces are operating in accordance with the ACS algorithm and design specifications. Though the exact on-orbit magnetic field cannot be simulated, the air bearing tests were designed to ensure that the magnetometers can sense a need for attitude adjustment. The measurements are compared to ephemeris magnetic field data to determine the amount of counter torque required.

This test was accomplished by measuring the magnetic field surrounding the air-bearing platform. The results were used as the input needed to establish ephemeris tables. Ephemeris magnetic field values, *B,* were compared to magnetometer readings. The difference between the two is considered the error between the actual field and the desired field. The error was used to determine the required amount of dipole moment ordered by the ACS. Moreover, once the onboard PC develops the attitude change commands, the air–bearing test allowed the actual witnessing of the ACS making a change based upon the sensed and processed data.

B. THESIS ORGANIZATION

This thesis is separated into six chapters including the introduction.

Chapter I is an introduction regarding NPSAT1, its mission, and thesis objectives.

Chapter II describes the overall NPSAT1 project to provide the reader with an introduction to the spacecraft that was studied. The description includes the satellite's generic structure, NPSAT1 constraints and design specifications, as well as the experiments being flown on the spacecraft.

Chapter III describes the mathematical theory referenced to derive the ACS program and model. Furthermore, Leonard's ACS program and model are discussed and compared against the program and model developed for comparison. The model is presented using both MATLAB and SIMULINK programs.

Chapters IV and V discuss the air–bearing laboratory.

Chapter V compiles and presents the results from the air-bearing tests.

Chapter VI contains conclusions regarding the validity of Leonard's attitude control model and program. The conclusions are summarized based on the computer simulation results and air–bearing tests. Moreover, further NPSAT1 ACS research is identified and discussed.

Appendix A contains the International Geomagnetic Reference Field coefficients used to calculate ephemeris data required by the ACS to model the Earth's magnetic signature.

Appendix B contains the ACS MATLAB program developed to verifiy and validate Leonard's work.

Appendix C holds the ACS MATLAB program developed by Leonard.

Appendix D displays a cross sectional view of a magnetic torque rod used on the

ACS air–bearing test platform.

Appendix E displays the SIMULINK sub–models built to verifity Leonard's ACS model.

Appendix F displays the SIMULINK model developed by Leonard.

Appendices G, H, and I contain the MATLAB programs used to model the magnetic field of the air–bearing test platform laboratory.

Appendix J displays the SIMULINK model used during the ACS air–bearing *hard-ware–in–the–loop* tests.

Appendix K contains the MATLAB program written that interfaces with the air–bearing control software and command computer.

Appendix L contains the MATLAB programs drafted to analyze the data collected while measureing the magnetic torque rod's induced magnetic fields and dipole moments.

II. NPSAT1 PROJECT DESCRIPTION

A. NPSAT1 SPECIFICATIONS

NPSAT1 is an experimental satellite under construction at NPS. The NPSAT1 mission was conceived and developed by the NPS SSAG. NPSAT1 is sponsored and executed by the DOD Space Test Program (SMC Det 12).

Three objectives were defined in support of NPSAT1. The first was to provide a platform for NPS officer–students to apply their lessons to a space system that will actually fly. The second goal was to provide a space platform that will carry six research experiments, the testing of an attitude control system (ACS) based solely on magnetic control, and the testing of new and unproven SPECTROLAB solar cells. The last goal was to support the NPS space systems, mechanical, and electrical engineering programs with a satellite that complements PANSAT (Petite Amateur Navy Satellite), the current NPS satellite in orbit. NPSAT1 incorporates the lessons learned from the PANSAT program while simultaneously exploring recent innovative techniques in small satellite construction that were not available during PANSAT construction.

The satellite will house six experiments. The onboard experiments include:

- Naval Research Laboratory (NRL)–sponsored Coherent Electromagnetic Radio TOmography (CERTO) test
- NRL–sponsored Langmuir Probe designed to perform on–orbit measurements of inonspheric Total–Electron Content (TEC).
- NPS–sponsored Configurable Processor Experiment (CPE) that tests the use of an adaptable processor using field-programmable gate array technology with non-volatile FERRO Random Access Memory (RAM).
- NPS–sponsored Commercial–Off–the–Shelf (COTS) visible-wavelength imager (VISIM) designed to produce less than one kilometer of optic resolution.
- NPS–sponsored three-axis Micro-Electromechanical Systems (MEMS) –based rate gyro data.
- NPS–sponsored Solar Cell Measurement System (SMS) demonstrations that include I–V measurements of two SPRECTROLAB advanced, triple–junction (ATJ) solar cells.

The satellite's scheduled launch is October 2006. The launch vehicle is a Lockheed Martin Atlas V rocket equipped with a medium Evolved Expendable Launch Vehicle (EELV) configured with a Secondary Payload Adapter (EPSA).

NPSAT1 is a prolate body that will fly in a circular Low Earth Orbit (LEO). It is a gravity-gradient friendly, nadir pointing, three–axis non-spinning body. NPSAT1 uses a command and data handling (C&DH) subsystem, an electrical power (EPS) subsystem, an attitude control (ACS) subsystem, and a radio frequency (RFS) subsystem.

The C&DH subsystem consists of a PC/104 bus, motherboard, mass storage, A/D conversion, input/output, power supplies, modem, and the configurable processor experiment (CPE). All C&DH components are housed in a rectangular box that can withstand the harsh space environment. A 386 microprocessor running at a rate of 33 MHz is used as a central processing unit. The power supply provides 3.3 V, 5.0 V, and 12.0 V DC. All electronics are constructed for improved electromagnetic compatibility. Using a Linux, POSIX–compliant operating system, the spacecraft is able to use software that is readily available and can be altered without additional cost. Specifically, the ACS software is written in MATLAB and SIMULINK. The code and model are converted to ANSI 'C' computer code using the SIMULINK Real Time Workshop toolbox, XPC Target and XPC Host functions.

The EPS subsystem consisting of SPECTROLAB triple junction solar cells are used to convert sunlight into electricity. The captured energy charges a lithium–ion (Li-ion) battery. The solar cells are mounted statically to the spacecraft chassis and possess 24% minimum efficiency.

The NPSAT1 attitude control system (ACS) is based upon the exploitation of the Earth's magnetic field and gravity–gradient friendly moments of inertia where $I_y > I_x > I_z$. Nearly symmetrical in all aspects, NPSAT1 possesses individual moments of inertia slightly larger than each other, so it takes advantage of the Earth's gravitational attraction. This concept assists in maintaining the spacecraft's desired attitude.

Three magnetic torque rods (solenoids) provide active control. A three-axis magnetometer senses the Earth's magnetic field at a predetermined moment in time. The measurement is compared against an onboard look–up table based upon current latitude, longitude, and altitude. This look–up table is a list of calculations that estimate the Earth's magnetic field. The

ephemeris data is based upon a 10^{th}-order, 10^{th}-degree spherical harmonic model. Calculations use the International Geomagnetic Reference Field (IGRF) coefficients as entering arguments. Each table value is corrected for the estimated value corresponding to a particular point in time and space. Based upon the tabulated and measured data, the ACS performs attitude corrections by applying current to three torque rods designed to produce a dipole that can either attract or repel. However, if the on–board magnetometer completes a measurement while the torque rods are conducting an actuation maneuver, the measurement received will be inaccurate because of electromagnetic interference. Thus, the magnetometer measures the field following the successful completion of torque rod excitation and full decay.

The radio frequency subsystem is comprised of a dual antenna system that points in both the nadir and zenith directions. An up-link frequency is 1767.565 MHz. The down-link frequency is tuned to 2207.3 MHz. A 100 kbps transmission is achievable using a full duplex channel. The ground station is located at the Naval Postgraduate School, Monterey, California, at a latitude of 36.6° N. The satellite will communicate with the ground station for approximately 34 minutes each day. NPSAT1 will pass over the ground station five times per day with each pass equal to approximately six minutes of communication.

NPSAT1 weighs 81.6 kg (180 lbs). The spacecraft is a prolate, 12-sided cylindrical body with solar cells mounted on each of its sides. Each end has an antenna mounted for communication purposes. The satellite will fly at an altitude of 600 ± 40 km with an inclination of 35.4 degrees. NPSAT1 is slightly gravity–gradient stabilized. Table 2.1 lists the satellite's Moments of Inertia (MOI), Center of Mass (*cm*) and Center of Pressure (*cp*), and the offset (*dL*) specifications with reference to the body axes. The MOI are the most recent values as of 22 August 2004 and are subject to change slightly. The MOI is measured in $\mathrm{kg} \cdot \mathrm{m}^2$. The offset, *dL*, is measured in meters.

I_{xx}	I_{xy}	I_{xz}	I_{yx}	I_{yy}	I_{yz}	I_{zx}	I_{zy}	I_{zz}	dL_x	dL_y	dL_z
5.404	–0.002	0.016	–0.002	5.558	–0.034	0.016	–0.034	2.416	–0.004	0.0002	0.389

Table 2.1 NPSAT1 Moments of Inertia & Center of Pressure – Center of Mass Offset [Provided by Mr. Dan Sakoda, NPS SSAG]

Figure 2.1 depicts the construction of NPSAT1. Figures 2.2 and 2.3 are computer-aided drafted depictions of the satellite.

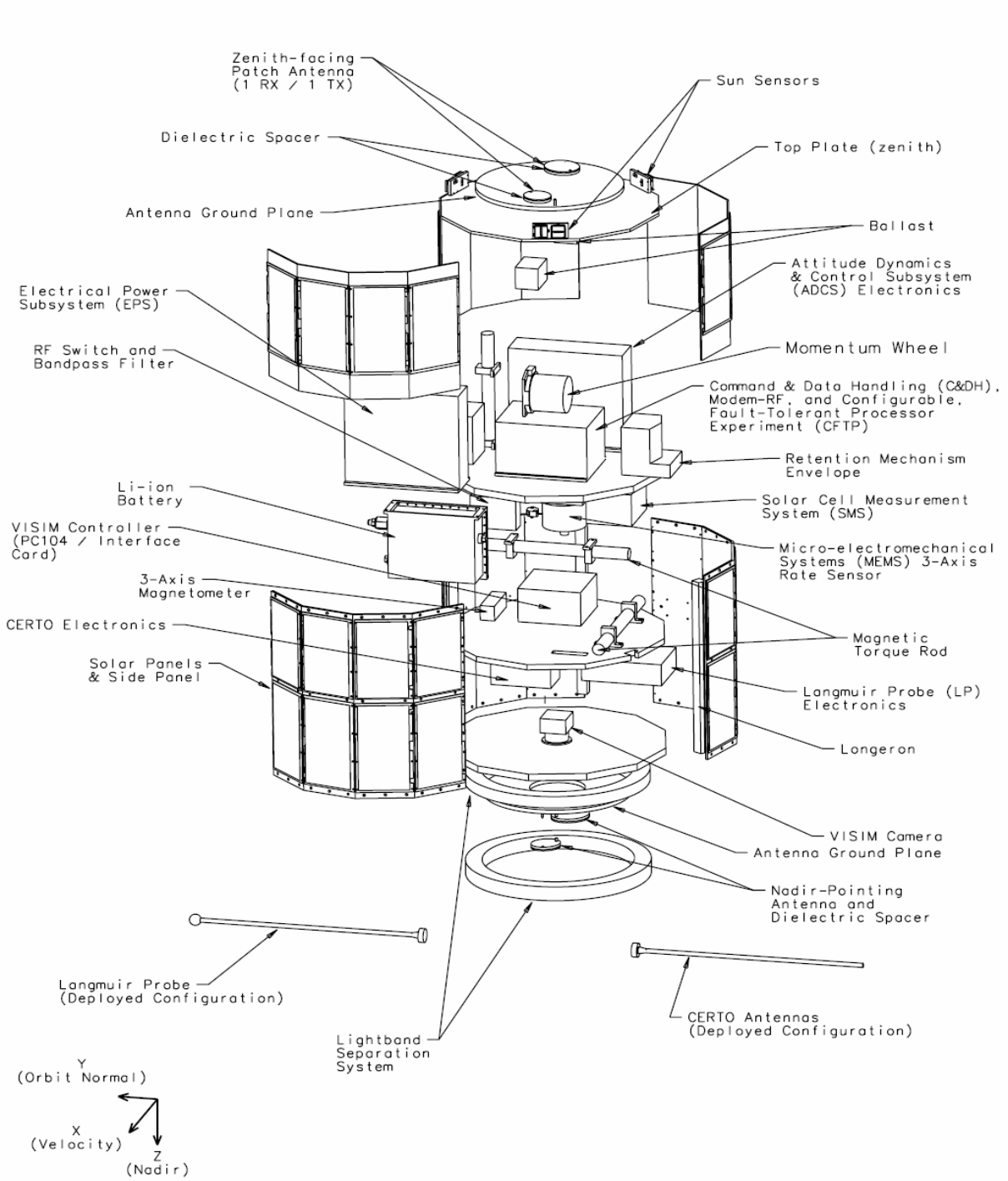


Figure 2.1 NPSAT1 Modular Component Model
[Provided by Mr. Dan Sakoda, NPS SSAG]

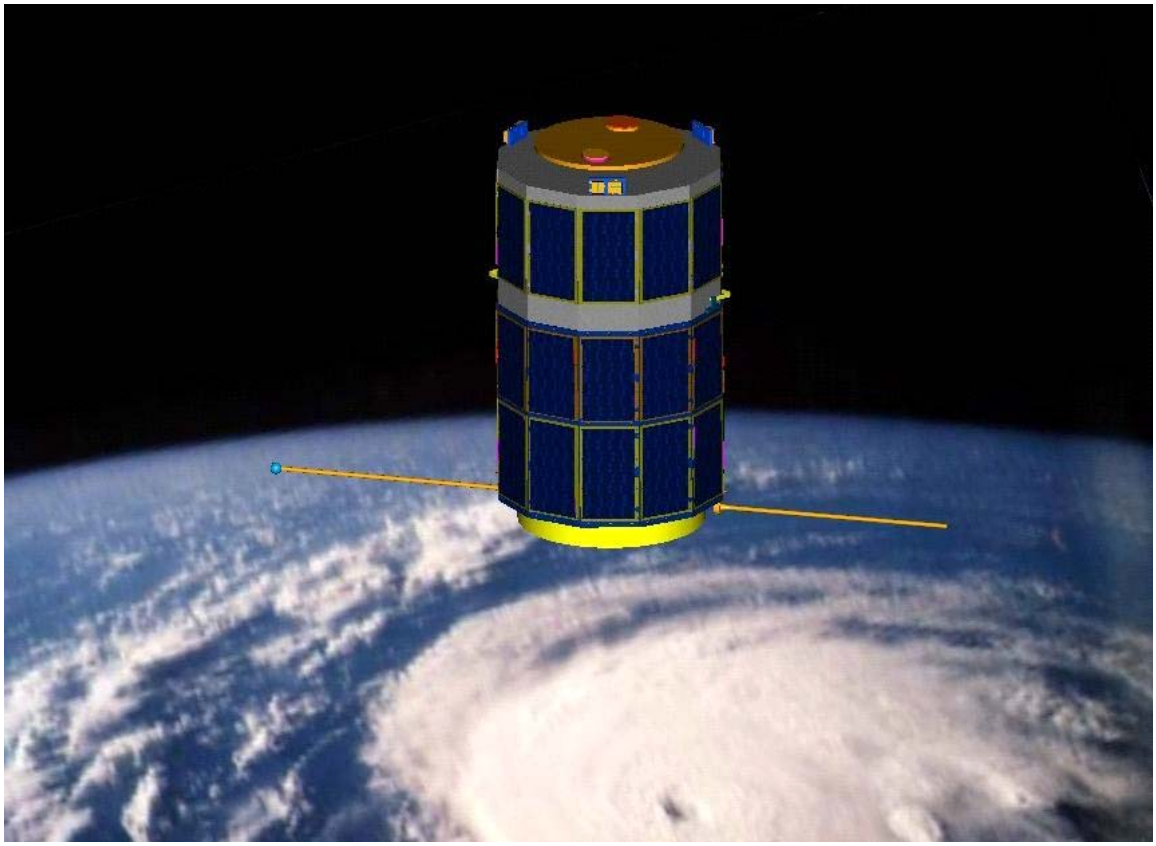


Figure 2.2 NPSAT1 Computer Generated Model
[Provided by Mr. Dan Sakoda, NPS SSAG]

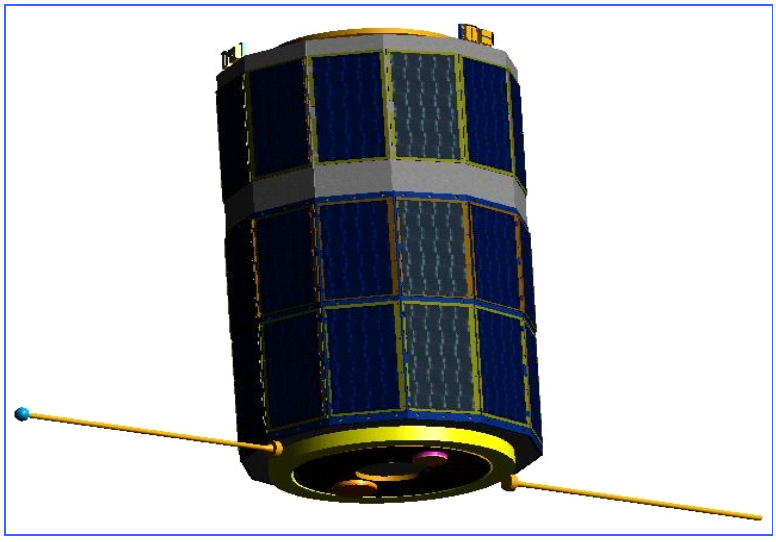


Figure 2.3 NPSAT1 Computer Model with Langmuir, CERTO Probe, Sun Sensors, Nadir Pointing Antennas and Solar Panels
[Provided by Mr. Dan Sakoda, NPS SSAG]

In summary, this chapter provided a description of NPSAT1, its mission, and design criteria. Chapter III will expand upon this information by discussing in detail the satellite's attitude control system and the scientific theory it is based upon.

III. NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM EMPERICAL DERIVATION, CONTROL LAWS, AND SIMULINK MODEL

A. INTRODUCTION

NPSAT1 is designed to change its attitude based upon measurement of the Earth's magnetic field, $\vec{B}$, which is compared to a table of theoretical values, $\vec{b}$. (Within this thesis vectors are represented as variables with a vector arrow displayed above. Matrices are noted as bolded or bracketed variables.) The theoretical values reside within the satellite's onboard computer. The attitude control model written in MATLAB and SIMULINK takes the cross product of $\vec{B}$ with $\vec{b}$ as an estimated attitude error. This error used to calculate the amount of torque required to adjust the satellite's attitude using a 3-2-1 Euler angle rotation sequence. The 3-2-1 rotational sequence corresponds to the rotational movement through the ψ, θ, ϕ Euler angles. Additionally, the model compensates for environmental disturbance torques produced by the gravity–gradient, the solar, and the aerodynamic forces. Therefore, this chapter discusses the mathematics and the control law theory that the ACS is derived from. Items addressed include the modeling of the Earth's magnetic field in a low Earth orbit, spacecraft kinematics and dynamics, disturbance torques, spacecraft tip–off attitude control, attitude control laws, and the ACS model.

B. EARTH'S MAGNETIC FIELD

"The Earth's magnetic field is predominantly that of a magnetic dipole such as that produced by a sphere of uniform magnetization or a current loop." [1, p. 113]. The strength of the field is weakest at the magnetic equator and increases exponentially as the magnetic latitude increases. As the distance from the Earth increases, the magnetic field strength decreases. This decrease is proportional to $1/r^3$ where r is the distance from the Earth's center of mass.

Spherical harmonics are used to calculate the theoretical magnetic field. The field is viewed as theoretical because it is in a constant state of flux. The scalar potential, $V(r,\theta,\phi)$, is given by [1, p. 779] as

$$V(r,\theta,\phi) = R_e \sum_{n=1}^{k} \left(\frac{R_e}{r}\right)^{n+1} \sum_{m=1}^{k} \left(g_n^m \cos m\phi + h_n^m \sin m\phi\right) P_n^m(\theta) \qquad (3.1)$$

which is the foundation for the calculations to determine the magnetic field, $\vec{B} = -\nabla V$. The magnetic field, $\vec{B}$, is negative of the gradient of the scalar potential, $V(r,\theta,\phi)$. [1] The symbol, R_e, is the Earth's equatorial radius, *r* is the distance from the center of Earth, the angle θ is the co–latitude and the angle ϕ is the east longitude measured from the Prime Meridian at Greenwich, England. The scalar, $P_n^m(\theta)$, is a Legrendre function. The last two variables, g_n^m and h_n^m are Gaussian coefficients. The sub– and super–scripts, *n* and *m*, represent the equation's degree and order, respectively.

A 10th-degree (*n*), 10th-order (*m*) model using Gaussian coefficients was used. Specifically, the coefficients are extracted from data calculated by International Association of Geomagnetism and Aeronomy (IAGA). The IAGA possess the most current International Geomagnetic Reference Field (IGRF) 2000 model that contains corrections up to 2005. Coefficient units are nano-Tesla (nT). A list of the IGRF coefficients used is provided in Appendix A. The following equations were used to calculate the three axes of the magnetic field in spherical harmonic coordinates. Further reference can be found in [1, p. 779].

Equation (3.2) performs a Gaussian normalization producing Schmidt factors ($S_{n,m}$). Here, $P_n^m = S_{n,m}P^{n,m}$ relates the Schmidt functions to Gaussian functions and is given by [1] as

$$S_{n,m} \equiv \left[\frac{\left(2-\delta_m^0\right)(n-m)!}{(n+m)!}\right]\frac{(2n-1)!}{(n-m)!}. \tag{3.2}$$

The Kronecker delta is $\delta_j^i = 1$ if $i = j$; otherwise it is 0. The *n* and *m* values relate to the IGRF coefficient order and degree indices as listed in Appendix A. The Schmidt factors are multiplied by the IGRF coefficients to produce values used in determining the individual components (r,θ,ϕ) of the magnetic fields, $\vec{B}$. Since recursive relationships are used, the Gaussian normalization is computed only once by

$$g^{n,m} \equiv S_{n,m}g_n^m, \quad h^{n,m} \equiv S_{n,m}h_n^m. \tag{3.3}$$

However, for computer operations, the Schmidt functions are further defined as [1]

$$\begin{aligned} S_{0,0} &= 1 \\ S_{n,0} &= S_{n-1,0}\left[\frac{(2n-1)}{n}\right] && \text{for } n \geq 1 \\ S_{n,m} &= S_{n,m-1}\sqrt{\frac{(n-m+1)\left(\delta_m^1+1\right)}{(n+m)}} && \text{for } m \geq 1 \end{aligned} \tag{3.4}$$

and the Gaussian functions are [1]

$$\begin{aligned} P^{0,0} &= 1 \\ P^{n,n} &= \sin(\theta)P^{n-1,n} \\ P^{n,m} &= \cos(\theta)P^{n-1,m} - K^{n,m}P^{n-2,m} \end{aligned} \tag{3.5}$$

where

$$k^{n,m} \equiv \begin{cases} (n-1)^2 & \text{for } n > 1 \\ 0 & \text{for } n = 1. \end{cases} \tag{3.6}$$

Since the magnitude of the magnetic field, B, is magnitude of the gradient of V, partial derivatives occur when performing Gauss function calculations. Therefore, the partial derivatives are [1]

$$\begin{aligned} \frac{\partial P^{0,0}}{\partial \theta} &= 0 \\ \frac{\partial P^{n,n}}{\partial \theta} &= \sin(\theta)\frac{\partial P^{n-1,n-1}}{\partial \theta} + \cos(\theta)P^{n-1,n-1} \\ \frac{\partial P^{n,m}}{\partial \theta} &= \cos(\theta)\frac{\partial P^{n-1,m}}{\partial \theta} - \sin(\theta)P^{n-1,m} - K^{n,m}\frac{\partial P^{n-2,m}}{\partial \theta}. \end{aligned} \tag{3.7}$$

The magnetic field components are calculated by [1]

$$\begin{aligned} B_r &= -\frac{\partial V}{\partial r} = \sum_{n=1}^{k}\left(\frac{R_e}{r}\right)^{n+2}(n+1)\sum_{m=0}^{n}\left(g^{n,m}\cos(m\phi) + h^{n,m}\sin(m\phi)\right)P^{n,m}(\theta) \\ B_\theta &= -\frac{1}{r}\frac{\partial V}{\partial r} = -\sum_{n=1}^{k}\left(\frac{R_e}{r}\right)^{n+2}\sum_{m=0}^{n}\left(g^{n,m}\cos(m\phi) + h^{n,m}\sin(m\phi)\right)\frac{P^{n,m}(\theta)}{\partial \theta} \\ B_\phi &= -\frac{1}{r\sin(\theta)}\frac{\partial V}{\partial r} = -\frac{1}{\sin(\theta)}\sum_{n=1}^{k}\left(\frac{R_e}{r}\right)^{n+2} \\ &\qquad \times \sum_{m=0}^{n} m\left(-g^{n,m}\sin(m\phi) + h^{n,m}\cos(m\phi)\right)P^{n,m}(\theta). \end{aligned} \tag{3.8}$$

The magnitude of the Earth's magnetic field, B, is the magnitude of the three component vectors [1]

$$B = \sqrt{B_r^2 + B_\theta^2 + B_\phi^2}. \tag{3.9}$$

Appendix B is the MATLAB program written to determine B using Equations (3.2–3.9).

Figure 3.1 is a projection using 2000 IGRF coefficients corrected to 2005 which was created using the Finnish Meteorological Institute IGRF Model Applet. [2]

Figure 3.2 is a computed contour plot of the Earth's magnetic field at 600 kilometers above the Earth's surface. The model uses 2000 IGRF coefficients corrected to 2005. It was developed from a program written in MATLAB code. As expected, the two plots are similar, thus proving that the Earth's magnetic field model is correct.

The external magnetic field values determined by the MATLAB program were preloaded in the ACS SIMULINK external magnetic field sub–model. Specifically, Figures E.12 and E.13 in Appendix E represent the stored magnetic field ephemeris data and the conversion from orbit–to–body coordinates. Sub–model entering arguments include the spacecraft's longitudinal position, Earth spin rate, nodal precession, inclination, and true anomaly.

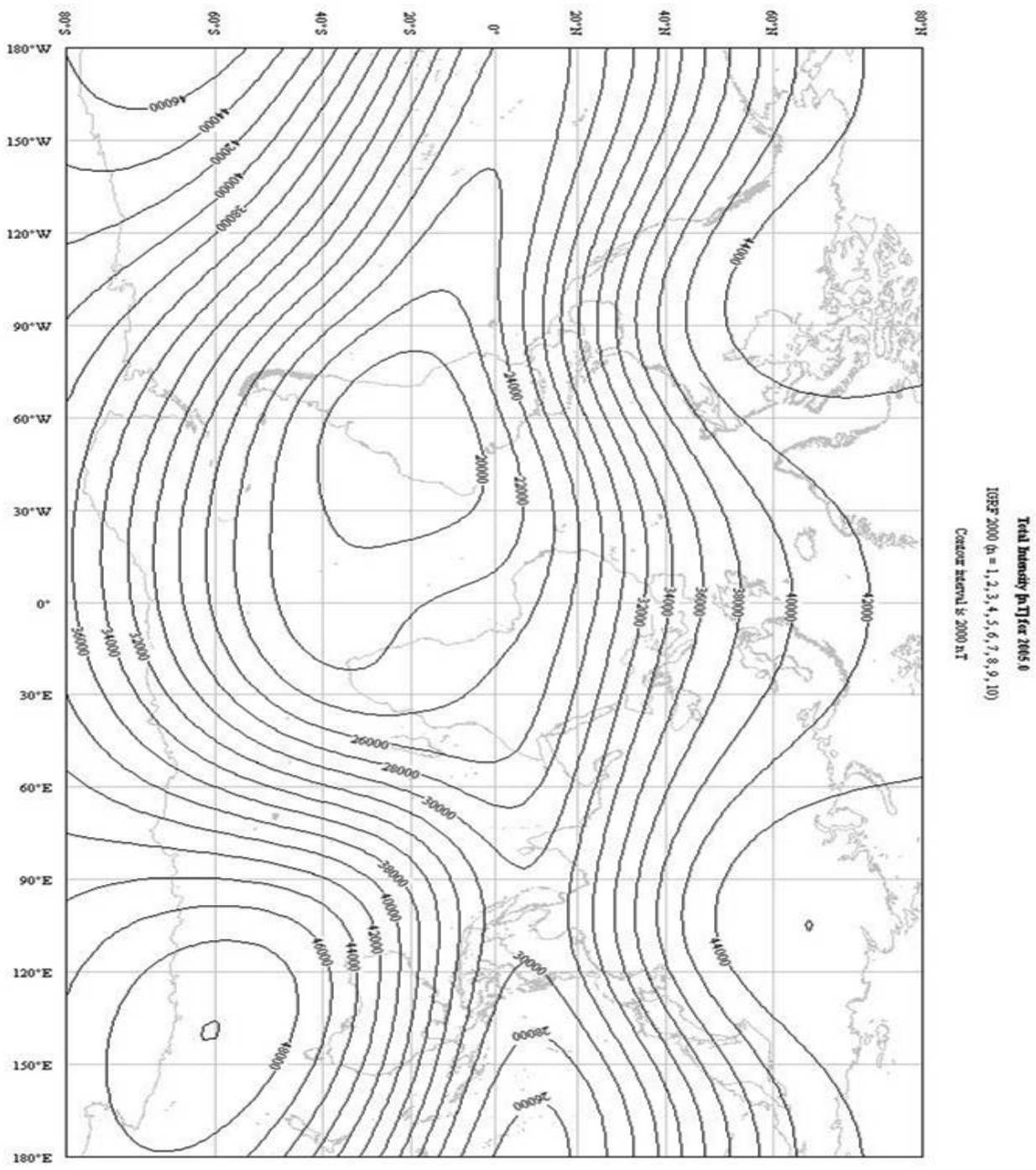


Figure 3.1 Finnish Meteorological Institute 2005 IGRF Earth Magnetic Field Contour Plot at 600 km above the Earth (2000 nano -Tesla contours) [From Ref. 2.]

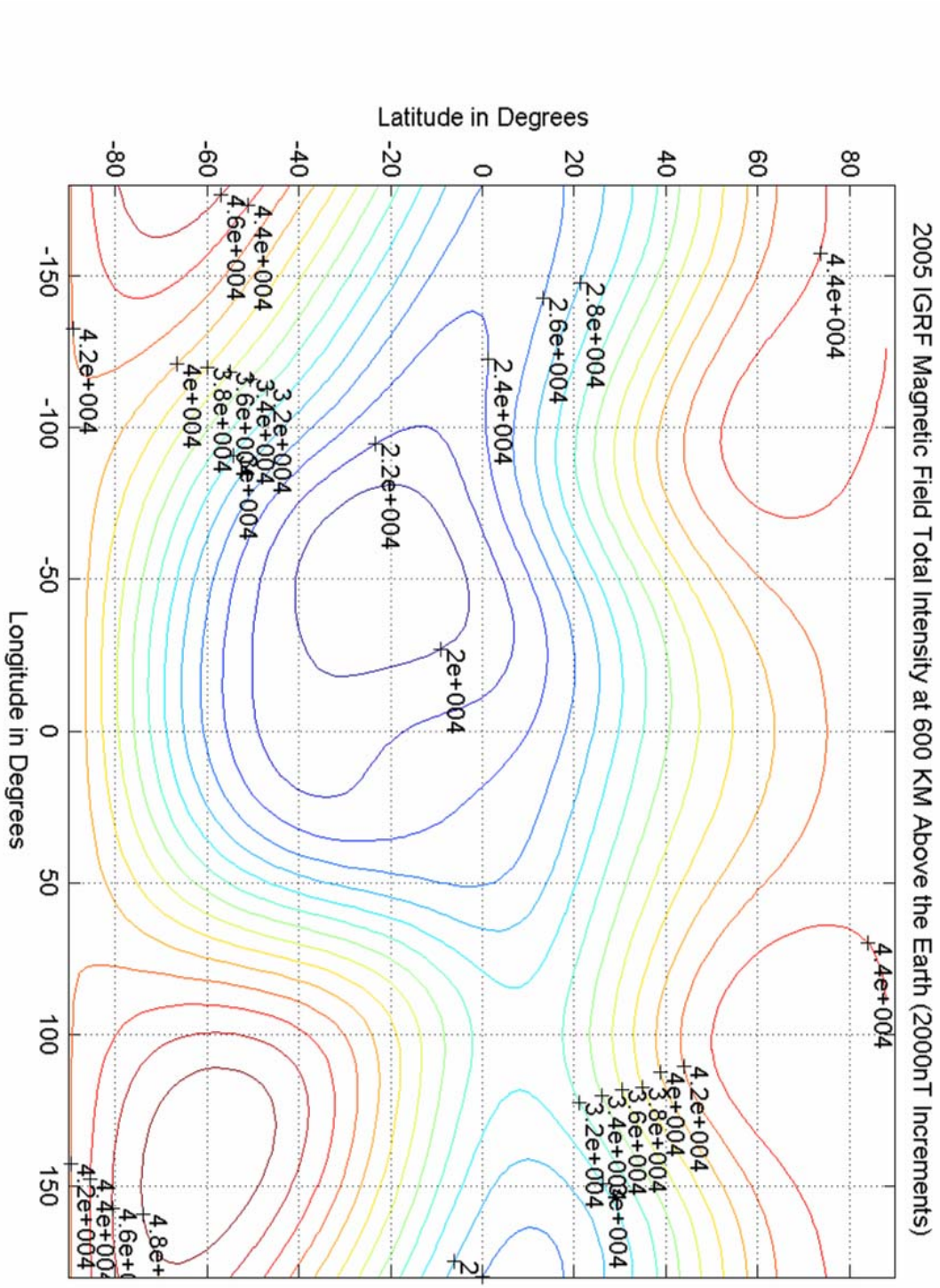


Figure 3.2 NPSAT1 MATLAB Generated Earth's Magnetic Field at 600 km

C. NPSAT1 KINEMATICS AND DYNAMICS

NPSAT1 uses an active three-axis magnetic attitude control system. Its rotation scheme is a 3-2-1 sequence and uses quaternions to express the direction cosine matrix (DCM). The 3-2-1 sequence relates to the order of rotation about a specific axis. Here, 3-2-1 refers to the angular rotation about the psi, theta, phi, (ψ,θ,ϕ) Euler angles. For a 3-2-1 system, Equation (3.10) represents the DCM, matrix **A,** desired where C = Cos and S = Sin. [3, p. 321]

$$\left[\mathbf{A}_{321}\right]=\left[\mathbf{A}_{\psi\theta\phi}\right]=\begin{bmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ -C\phi S\psi + S\phi S\theta C\psi & C\phi C\psi + S\phi S\theta S\psi & S\phi C\theta \\ S\phi S\psi + C\phi S\theta C\psi & -S\phi C\psi + C\phi S\theta S\psi & C\phi C\theta \end{bmatrix} \tag{3.10}$$

However, to prevent singularities, the elements of the quaternion vector, $\vec{q}$, are used and are related to the DCM by Equations (3.11), (3.12), and (3.13) [4, p. 27]

$$\begin{aligned} \vec{q} &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \\ q_1 &= S_1C_2C_3 - C_1S_2S_3 \\ q_2 &= C_1S_2C_3 - S_1C_2S_3 \\ q_3 &= C_1C_2S_3 - S_1S_2C_3 \\ q_4 &= C_1C_2C_3 - S_1S_2S_3 \end{aligned} \tag{3.11}$$

where "1" is $\phi/2$, "2" is $\theta/2$, and "3" is $\psi/2$. The DCM, ${}^{O}\mathbf{C}^{B}$, using quaternion values in the orbit-to-body frame reference plane is [4, p. 27]

$${}^{O}\mathbf{C}^{B}=\left[\mathbf{C}_{ij}\right]=\begin{bmatrix} q_1^2-q_2^2-q_3^2+q_4^2 & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_1q_2-q_3q_4) & q_2^2-q_1^2-q_3^2+q_4^2 & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & q_3^2-q_1^2-q_2^2+q_4^2 \end{bmatrix}. \tag{3.12}$$

Thus, for a standard 3-2-1 sequence, the angular values are found using

$$\begin{aligned} \psi &= \tan^{-1}\left(\frac{C_{12}}{C_{11}}\right), \\ \theta &= -\sin^{-1}\left(C_{13}\right), \\ \phi &= \tan^{-1}\left(\frac{C_{23}}{C_{33}}\right). \end{aligned} \tag{3.13}$$

For a control system to adjust a satellite's attitude, it uses two other reference frames. Specifically, these are the body–to–inertial, ${}^{N}\vec{\omega}^{B}$, and orbit–to–inertial frames, ${}^{N}\vec{\omega}^{O}$. For this thesis, the inertial frame is designated as the Earth–bound frame. In the orbit frame, the *x*-axis is the spacecraft velocity vector, the *y*-axis follows the right-hand rule and the *z*-axis is nadir pointing. The angular velocity in each frame can be measured using these three axes reference frames. Furthermore, the body–to–orbit angular velocity, ${}^{O}\vec{\omega}^{B}$, can be computed using

$$ {}^{O}\vec{\omega}^{B} = {}^{N}\vec{\omega}^{B} - {}^{N}\vec{\omega}^{O} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}. \tag{3.14} $$

Additionally, the elemental values of ${}^{O}\vec{\omega}^{B}$ can be written in terms of the quaternion derivative as [4, p.26]

$$ \begin{aligned} \omega_1 &= 2\left(\dot{q}_1 q_4 + \dot{q}_2 q_3 - \dot{q}_3 q_2 - \dot{q}_4 q_1\right) \\ \omega_2 &= 2\left(\dot{q}_2 q_4 + \dot{q}_3 q_1 - \dot{q}_1 q_3 - \dot{q}_4 q_2\right) \\ \omega_3 &= 2\left(\dot{q}_3 q_4 + \dot{q}_1 q_2 - \dot{q}_2 q_1 - \dot{q}_4 q_3\right). \end{aligned} \tag{3.15} $$

State space representation of state vectors takes Equation (3.15) and develops it into a $\dot{x} = Ax + Bu$ format. [4] Specifically, the quaternion derivatives can be expressed in matrix format and in terms of the quaternion and angular velocity components as [4, p. 26]

$$ \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ \omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{3.16} $$

Here, ${}^{N}\vec{\omega}^{B}$ is determined from the spacecraft's initial angular velocity, ω_0 and feedback from the DCM. Specifically, ${}^{N}\vec{\omega}^{B}$ is found by multiplying ω_0 by the center column vector of the DCM or C_{i2}. This now leaves ${}^{N}\vec{\omega}^{B}$ to be found and is represented as $\begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. The dynamics function with Euler's spacecraft moment equations completes this relationship. Entering arguments include the disturbance torque, T_d; the control torque, T_c; the moments of inertia matrix (MOI), ***I***; the angular momentum, ***H***; and feedback from $\begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. The moments of inertia were listed in Table 2.1. The angular momentum is found through the inner product of the ***I*** matrix with the ${}^{N}\vec{\omega}^{B}$ vector or $\mathbf{H} = \mathbf{I} \cdot {}^{N}\vec{\omega}^{B}$. [4, p. 37] Therefore,

${}^{N}\vec{\omega}^{B} = [\mathbf{I}]^{-1}[\mathbf{H}]$. However, to determine the body–to–inertial angular velocity, the total torque acting on the satellite must be addressed.

Three types of external disturbance torques are applicable to the NPSAT1. These are the gravity–gradient torque, the solar torque, and the aero torque. Discussion regarding the calculation of the disturbance torques is reserved for a later section. However, when addressing torques within the ACS dynamics function, torque is defined as $T = T_c + T_d$. [3, p. 107] The Euler Moment Equations denote the individual torque relationships as

$$\begin{aligned} T_x &= I_{xx}\dot{\omega}_x + \left(I_{zz} - I_{yy}\right)\omega_y\omega_z \\ T_y &= I_{yy}\dot{\omega}_y + \left(I_{xx} - I_{zz}\right)\omega_z\omega_x \\ T_z &= I_{zz}\dot{\omega}_z + \left(I_{yy} - I_{xx}\right)\omega_x\omega_y. \end{aligned} \tag{3.17}$$

Furthermore, the total torque can be expressed in terms of the angular momentum, ***H*,** the rate of angular momentum change in the inertial frame, $\frac{d^N\mathbf{H}}{dt}$, and the body–to–inertial angular velocity, ${}^{N}\vec{\omega}^{B}$. Thus, ***T*** can be written [3]

$$\mathbf{T} = \dot{\mathbf{H}} = \frac{d^N\mathbf{H}}{dt} = \frac{d^B\mathbf{H}}{dt} + {}^{N}\vec{\omega}^{B} \times \mathbf{H} \tag{3.18}$$

and the rate of angular momentum change in the body frame is given by [3]

$$\frac{d^B\mathbf{H}}{dt} = \mathbf{T} - {}^{N}\vec{\omega}^{B} \times \mathbf{H}. \tag{3.19}$$

Lastly, the angular velocities vector with respect to the unit direction vectors $\begin{bmatrix}\omega_1 & \omega_2 & \omega_3\end{bmatrix}^T$ is determined by [3]

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix} = \begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} - \begin{bmatrix}0 & C_{12} & 0\\ 0 & C_{22} & 0\\ 0 & C_{32} & 0\end{bmatrix}\begin{bmatrix}0\\ -\omega_0\\ 0\end{bmatrix} \tag{3.20}$$

where ω_0 is identified as the angular orbital rate.

Figures E.4, E.5, and E.6 display the methods in which the spacecraft kinematics and dynamics were modeled. Specifically, entering arguments to the dynamics sub–model are the external disturbance torques (solar, aero, and gravity–gradient) and the ordered torques which are determined by the torque rod control law. The kinematics sub-model, Figure E.5, determines the body–to–orbit angular velocity based upon Equation (3.14). The sub–model is a

closed–loop feedback system designed to determine the quaternion vector using the quaternion derivatives as expressed in Equation (3.16). The output of the kinematics sub–model is the input to the direction cosine matrix sub–model. The DCM's output is the entering argument to the orbit–to–body coordinate conversion process. Additionally, the DCM also supplies feedback for the orbit–to–body angular velocity vector which is used to determine the body–to–orbit reference frame coordinates as expressed in Equation (3.14). The last function of the DCM is to provide input into the Sun and Eclipse sub–model. This sub–model is shown in Figure E.7 and is designed to compensate for the differences exhibited in external disturbance torques while the spacecraft is in eclipse.

D. NPSAT1 DISTURBANCE TORQUES

1. Gravity–Gradient Torque

"Inherent in low–orbit satellites … an asymmetric body subject to a gravitational field will experience a torque tending to align the axis of least inertia with the field direction." [3, p. 108] Since NPSAT1 is gravity–gradient friendly, the *z*-axis is the nadir–pointing vector. Moreover, in the calculation of the gravity–gradient torque, the moments of inertia for all three axes must be taken into account using

$$\begin{bmatrix} T_{GGx} \\ T_{GGy} \\ T_{GGz} \end{bmatrix} = \frac{3\mu}{R_0} \begin{bmatrix} C_{13} \\ C_{23} \\ C_{33} \end{bmatrix} \mathrm{X} \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} C_{13} \\ C_{23} \\ C_{33} \end{bmatrix}. \tag{3.21}$$

This represents the determination in body coordinates. The variable μ is the Earth's gravitational constant, $3.98601\times10^{14}\ \mathrm{m}^3/\mathrm{s}^2$. Here, R_0 equals the Earth's equatorial radius plus the spacecraft's height above the Earth surface, in meters. While the gravity–gradient torque is calculated using Equation (3.20). The spacecraft angular orbital velocity is determined by $\omega_0 = \sqrt{\mu / R_0^3}$.

The gravity–gradient disturbance torque sub–model is shown in Figure E.3. The gravity–gradient torque is determined by the cross product as expressed in Equation (3.20). Entering arguments are the DCM's third column and moment of spacecraft MOI values loaded from the ACS MATLAB .m file.

2. Aerodynamic Torque

NPSAT1 is designed to fly at an altitude of 600 ± 40 km in a Low Earth Orbit (LEO) with an eccentricity of zero. As explained by Wertz, "altitudes between 120 and 600 km are within the Earth's *thermosphere*…where absorption of extreme ultraviolet radiation from the Sun results in a very rapid increase in temperature with altitude". [1, p. 208]

The temperature fluctuates greatly as the satellite flies through different solar regions of a LEO. Accordingly, the atmospheric density is proportional to the temperature differences. Therefore, atmospheric drag will cause pressure upon the spacecraft body which disturbs its attitude positioning. Approximating the aerodynamic moment, [5] identifies

$$T_a = \left(c_p - c_m\right) F_{aero} \tag{3.22}$$

where

$$F_{aero} = \frac{1}{2} \rho C_d A V^2 \tag{3.23}$$

and

c_p = center of aerodynamic pressure,

c_m = center of mass,

$C_d = 2.5$,

A = spacecraft projected area,

ρ = atmospheric density, and

V = spacecraft velocity = $\sqrt{\mu / R_0}$,

as the governing equations used to calculate aerodynamic disturbances.

The atmospheric density has been studied and measured for many years. For the NPSAT1 model, values for ρ are those listed in [5]. The drag coefficient is C_d with values normally between 2 to 2.5. The NPSAT1 model uses the most conservative value of 2.5 for simulation purposes.

3. Solar Torque

Three factors regarding solar radiation and its accompanying torque are analyzed. These are spacecraft shape, Sun vector orientation with respect to the spacecraft, and solar intensity. Solar rays come directly from the Sun or are those that originate from the Sun, but are

reflected by the Earth. Another type spawns from radiation emitted by the Earth. Therefore, according to Wertz, "solar torque on a spacecraft is dependent upon the vehicle's geometry, surface reflectivity and center of mass location." [5, p. 366]

Modeling solar torque for the most conservative approximation is completed using [5]

$$T_{sp} = F\left(c_{ps} - c_m\right) \tag{3.24}$$

where

$$F = \frac{F_s}{c} A_s \left(1+q\right)\cos i \tag{3.25}$$

and

F_s = solar constant = $1{,}367\ \mathrm{W/m^2}$,
c = speed of light = 3.0×10^8 m/s,
A_s = spacecraft illuminated surface area,
c_{ps} = center of solar pressure,
c_m = center of mass,
q = reflectance factor (0 to 1), and
i = angle of incidence to the Sun.

However, for modeling the worst scenario possible, the reflectance factor is set to 1 and $P = 2F_s/c$ is used. The value calculated and used for NPSAT1 modeling was $2*\left(4.5\times10^{-6}\right)\mathrm{N/m^2}$.

The solar and aerodynamic disturbance torques were modeled as shown in Figure E.2. Since the aero torque is dependent upon the spacecraft's altitude, a look–up table was defined by values entered via the ACS MATLAB program. Additionally, the model takes into consideration the density variation seen while NPSAT1 is in eclipse and when it is not. The magnitude of the aero torque is governed by Equation (3.22). Furthermore, the solar torque is a direct calculation based upon Equation (3.24).

E. MAGNETIC ATTITUDE CONTROL

Torque can be produced based upon Earth's magnetic field and a magnetic moment. This control law is defined by [3]

$$\vec{T}_p = \overrightarrow{M} \times \vec{B} \tag{3.26}$$

where

$\vec{T}_p =$ torque acting on spacecraft,

$\vec{M} =$ spacecraft magnetic moment, and

$\vec{B} =$ Earth magnetic field.

A magnetic moment is created through the use of torque rods. The control equations are [3, p. 156] where the K_i component scalers and are selected through experimentation

$$\begin{aligned} T_{cs} &= 2K_x q_1 q_4 + K_{xd}\omega_1 \\ T_{cy} &= 2K_y q_2 q_4 + K_{yd}\omega_2 \\ T_{cz} &= 2K_z q_3 q_4 + K_{zd}\omega_3. \end{aligned} \tag{3.27}$$

The NPSAT1 attitude control system builds upon this theory. Specifically, Leonard [6] identifies a magnetic control law that follows the ideal quaternion control law of

$$u = K_a y + K_b \omega. \tag{3.28}$$

Comparing Equation (3.27) with Equation (3.28), a quaternion approximation can be completed through the use of measured magnetic field values crossed with ephemeris values. Specifically, Leonard proposes using an approximation for $\vec{y}$ based upon the measured and stored magnetic field values. [6]

Identified as the "Cross Product Steering Law" [6, p. 2], the estimation is the basis for developing a state vector, $\vec{y}$, which is used in the development of a Reduced Order Estimator (ROE).

$$\vec{y} = \left(\vec{B} \times \vec{b}\right) \Big/ \left|B^2\right| \approx 2\begin{bmatrix} q_1 q_2 & q_2 q_4 & q_3 q_4 \end{bmatrix}^T. \tag{3.29}$$

The $\vec{y}$ state vector is determined by the $\vec{B}$ and $\vec{b}$ magnetometer and ephemeris respective values. Additionally, the approximation is directly related to the quaternion vector.

Moreover, since NPSAT1 is gravity–gradient friendly, the desired magnetic damping moment can be achieved through

$$\overrightarrow{m_r} = \left(\vec{B} \times \vec{T}_r\right) \Big/ \left|B^2\right|. \tag{3.30}$$

This provides active damping to all axes. Instead of using Sidi's notation [3], T_c, to denote the control torque required, Equations (3.26) and (3.30) maintain the notation that is represented in [6]. Therefore, $\vec{T}_r$ and $\vec{T}_p$ represent the requested torque and produced torque, respectively.

Furthermore, "accuracies of the order of 2° are achievable with this technique". [3, p. 129] This degree of accuracy is well within the desired ± 10° pointing requirement for NPSAT1.

Following the actuator control law implementation, the amount of magnetic moment, $\vec{m}_r$, torque produced on the spacecraft is calculated using [3]

$$\vec{T}_p = \vec{m}_p \times \boldsymbol{\beta} \tag{3.31}$$

where combining the above, it can be shown that $\vec{T}_p = \boldsymbol{\beta}\vec{T}_r$ where the $\boldsymbol{\beta}$ matrix used in Equation (3.31) is [6]

$$\boldsymbol{\beta} = \begin{bmatrix} \frac{\left(B_y^2 + B_z^2\right)}{B^2} & \frac{-B_x B_y}{B^2} & \frac{-B_x B_z}{B^2} \\ \frac{-B_x B_y}{B^2} & \frac{\left(B_x^2 + B_z^2\right)}{B^2} & \frac{-B_y B_z}{B^2} \\ \frac{-B_x B_z}{B^2} & \frac{-B_y B_z}{B^2} & \frac{\left(B_x^2 + B_y^2\right)}{B^2} \end{bmatrix} \tag{3.32}$$

and is developed from data extracted from the IGRF calculations.

As [6] explains which cites [7], "The off diagonal terms of $\boldsymbol{\beta}$ have an average value of zero". [6, p. 3]. Furthermore, "the diagonal terms, defined as *gx*, *gy,* and *gz* vector components respectively, have average values that are a function of orbit inclination. Multiplying the components $\vec{T}_r$ of by reciprocals of *gx*, *gy,* and *gz*, respectively, yields an average value of $\vec{T}_p$ equal to $\vec{T}_r$." [6, p. 3] The values for *gx*, *gy,* and *gz* are produced within the MATLAB code located in Appendix B and C. Specifically, Figure E.10 displays the manner in which the diagonal terms are routed through the model.

Figure 3.3 is a block diagram model of the magnetic attitude control system. The figure displays the application of Equations (3.23–3.29) with an addition of a term called *Bdot* [6]. The *Bdot* term is used in the ACS system following tip-off from the EELV. Its execution provides initial satellite stabilization relative to the $\vec{B}$ vector following launch. Though desired pointing requirements will not be achieved using this function, a rough approximation results, thus allowing the Cross Product Steering Law to begin its damping of the satellite's pendulum motion.

Upon tip–off, it is assumed NPSAT1 will tumble; however, communication to the satellite is achieved via the dual antenna array. Therefore, positional data, specifically longitudinal coordinates, can be received. The longitudinal coordinates will be supplied from NORAD to the NPSAT1 ground station control. These are transmitted to NPSAT1 upon initialization.

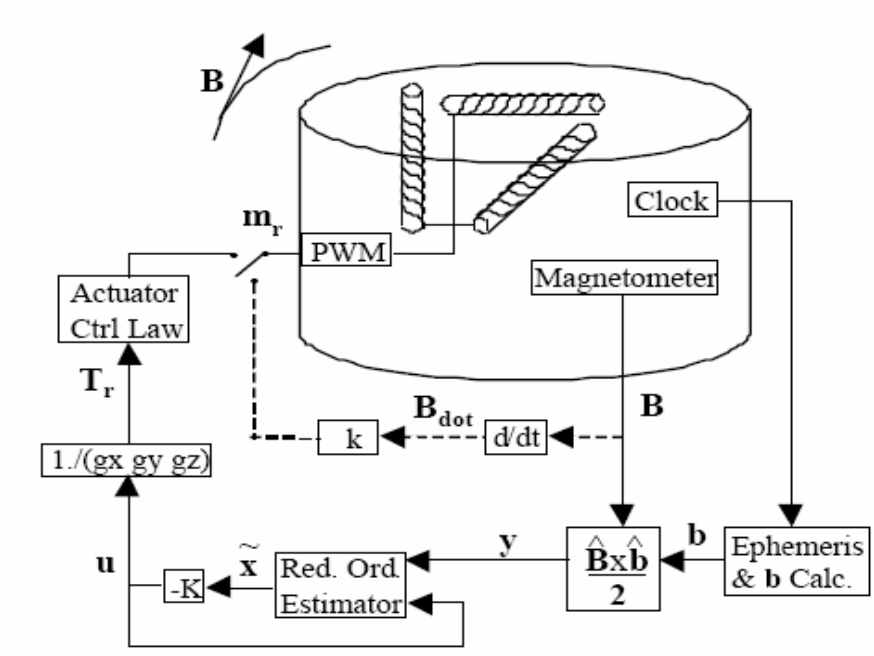


Figure 3.3 NPSAT1 3-Axis Control Using Magnetometers and Torque Rods [From Ref. 4.]

F. MAGNETIC TORQUE RODS AND CONTROL

NPSAT1 utilizes three magnetic torque rods (solenoids) produced by MICROCOSM. Specifically, NPSAT1 was initially designed to carry three model MT-30-2-CGS torque rods. Table 3.1 identifies the torque rod specifications.

Appendix D is a cross sectional view of a NPSAT1 torque rod. The flight torque rods were used during ACS simulation on an air bearing test platform. Tables 3.2–3.4 list magnetic torque rod specifications provided by MIRCROCOSM. Each table represents an acceptance test that determines the amount of magnetic moment created by a specific drive current. The

tests were completed using two current settings, ±10 mA and ±145 mA . Figure 3.4 is a picture of a magnetic torque rod that was to fly on NPSAT1.

Each rod consists of two sets of coils; however, only one coil will be driven at a time. Coil activation is determined by the type of magnetic moment required as calculated by the ACS program.

Type	**Linear Dipole Moment [A·m^2]**	**Saturation Moment [A·m^2]**	**Length [mm]**	**Diameter [mm]**	**Mass [kg]**	**Power [W]**	**Voltage [V]**	**Status**
MT-30-2-CGS	30	40	351	27	1.135	3.6	25	In Orbit

Table 3.1 NPSAT1 MICROCOSM Magnetic Torque Rod Specifications [After Ref. 8.]

Coil A	**Current**	–10 mA	+10 mA	–145 mA	+145 mA
	Dipole	–2.20 A·m^2	+2.40 A·m^2	–33.0 A·m^2	33.1 A·m^2
	Slope	0.230 A·m^2/mA	0.230 A·m^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA
Coil B	**Current**	–10 mA	+10 mA	-14 5 mA	+145 mA
	Dipole	–2.30 A·m^2	+2.30 A·m^2	–33.0 A·m^2	33.0 A·m^2
	Slope	0.230 A·m^2/mA	0.230 A·m^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA

Table 3.2 NPSAT1 Torque Rod Magnetic Moment vs. Drive Current (X-axis) [8]

Coil A	**Current**	–10 mA	+10 mA	–145 mA	+145 mA
	Dipole	–2.30 A·m^2	+2.25 A·m^2	–33.05 A·m^2	33.0 A·m^2
	Slope	0.227 Am^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA	0.227A·m^2/mA
Coil B	**Current**	–10 mA	+10 mA	–145 mA	+145 mA
	Dipole	–2.35 A·m^2	+2.3 0A·m^2	–33.1 A·m^2	33.0 A·m^2
	Slope	0.232 A·m^2/mA	0.232 A·m^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA

Table 3.3 NPSAT1 Torque Rod Magnetic Moment vs. Drive Current (Y-axis) [8]

Coil A	**Current**	–10 mA	+10 mA	–145 mA	+145 mA
	Dipole	–2.40 A·m^2	+2.25 A·m^2	–33.1 A·m^2	33.0 A·m^2
	Slope	0.232 A·m^2/mA	0.232 A·m^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA
Coil B	**Current**	–10 mA	+10 mA	–145 mA	+145 mA
	Dipole	–2.35 A·m^2	+2.25 A·m^2	–33.1 A·m^2	33.0 A·m^2
	Slope	0.230 A·m^2/mA	0.230 A·m^2/mA	0.227 A·m^2/mA	0.227 A·m^2/mA

Table 3.4 NPSAT1 Torque Rod Magnetic Moment vs. Drive Current (Z-axis) [8]

Figure 3.4 NPSAT1 MICROCOSM Torque Rods [From Ref. 8.]

As previously stated, the NPSAT1 attitude control is commanded by software drafted using MATLAB and SIMULINK. Figure 3.5 is a pictorial representation of the SIMULINK model used to drive the attitude control system. The model includes all disturbance torques, ephemeris data, the cross product steering law, a reduced order estimator feedback loop, and the torque rod actuator control function. Further discussion regarding the model is reserved for Chapter III, Section G and Chapters IV and V.

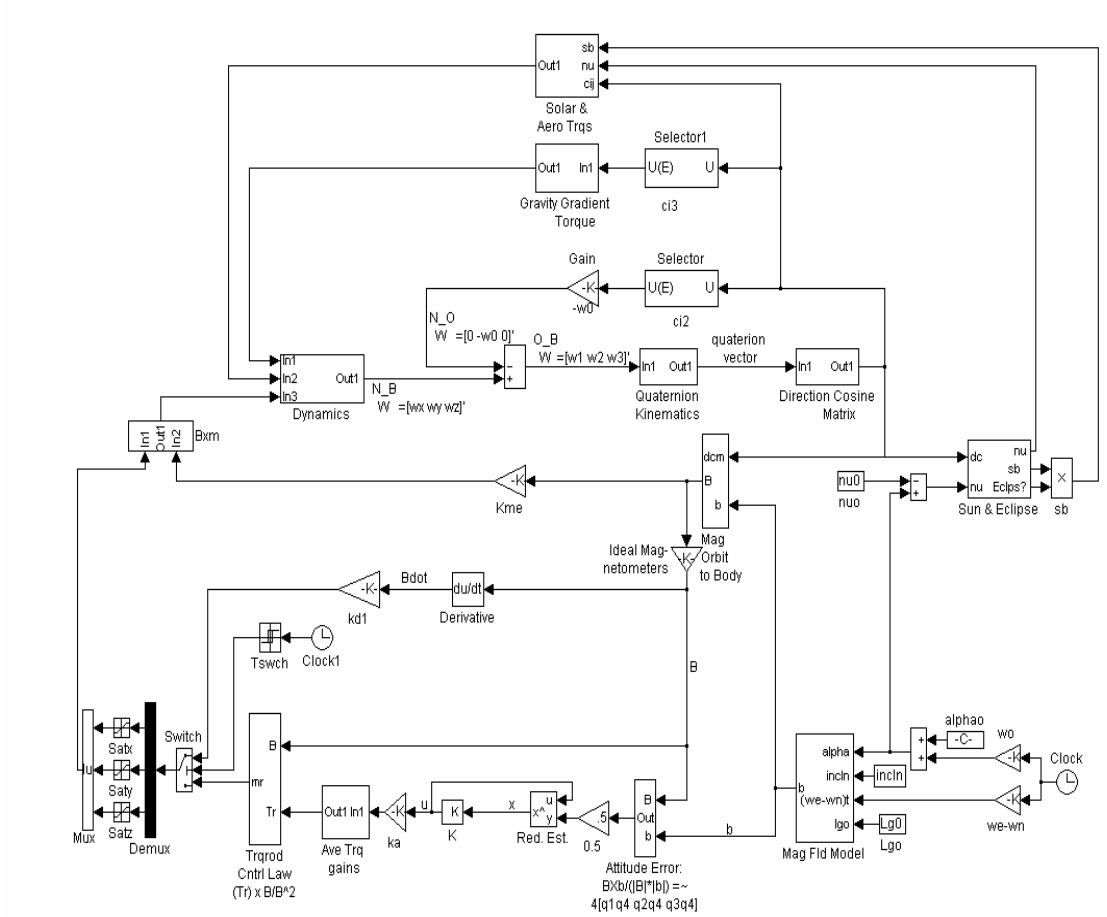


Figure 3.5 NPSAT1 ACS SIMULINK Model H with Disturbances and Ephemeris Input [After Ref. 9.]

Specifically, the ACS MATLAB code and SIMULINK models were converted and compiled into ANSI 'C' code format using the SIMULINK Real Time Workshop tool. The ANSI 'C' code program was hosted onboard the NPSAT1 computer. The computer receives the incoming magnetometer readings, stores the ephemeris data, and computes the necessary torque commands, as shown in Figure 3.6. As Figure 3.6 demonstrates, the 3-axis magnetometer readings are sent to the onboard micro–controller that converts the values to Tesla. The REO input is an approximation of an attitude error vector, $\vec{y} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$. The REO estimates $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ to form the complete state vector, $\vec{x} = \begin{bmatrix} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$. It then multiplies this by an optimal gain $K \in \mathbb{R}^{3\times 6}$ to form the 3-axis control torque $\vec{u} = -K\vec{x}$. These are used to enter the SIMULINK model to perform the ROE calculations. If any difference

between the magnetometer readings and ephemeris data exists, the control signal is forwarded via the microcontroller to the torque rod driver board. This driver manages the current flow to the torque rod windings. The current–induced windings create a magnetic moment that reacts with the external magnetic field. Moreover, the magnetometer only measures the external magnetic field when the torque rods are non-operational. This is accomplished to prevent any warping of the magnetic field from the torque rod's dipoles, thus altering the control torque commands.

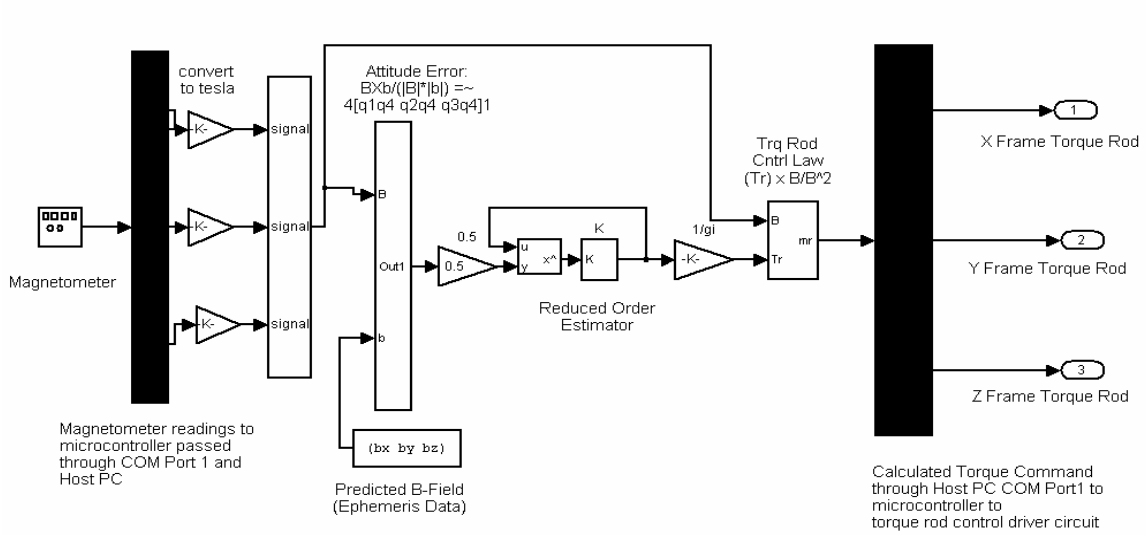


Figure 3.6 NPSAT1 Torque Rod Command Model [After Ref. 9]

Figure 3.7 is a portion of the SIMULINK model that is converted from a MAT-LAB/SIMULINK format to ANSI 'C' code using the SIMULINK Real Time Workshop function. This model is used to demonstrate the effectiveness of the ACS on the air-bearing platform. The input/output (I/O) device is a RS232 interface that receives magnetometer readings and sends a command to the torque rod driver board. The RS232 is the interface between the onboard computer microcontroller, SIMULINK model, host and target PCs, and torque rods.

Further discussion regarding the model is reserved for Chapter V.

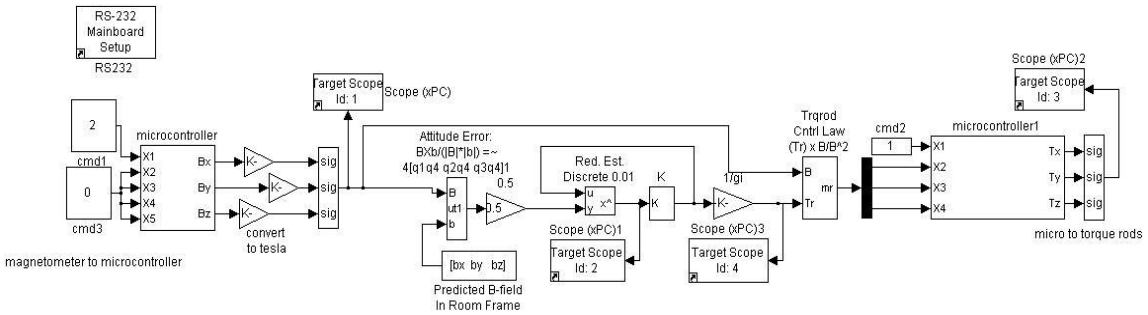


Figure 3.7 NPSAT1 ACS Air–Bearing SIMULINK Test Model

G. NPSAT1 MAGNETIC ATTITUDE CONTROL PROGRAM AND MODEL CONSTRUCTION

The MATLAB program constructed to control the spacecraft's magnetic moment via torque rods was designed in three sections.

First, as discussed in Chapter III, Section B, is the Earth's Magnetic Field program. This program was designed to calculate the ephemeris data required by the ACS plant.

Second is the establishment of the state space matrices and vectors required by the reduced order estimator, actuator control law, *Bdot* control law, cross product steering control law, disturbance torques, and spacecraft kinematics and dynamics.

Last is the development of the model for simulation and code generation purposes. Program sections one and two were drafted in MATLAB code. Section three was developed using SIMULINK modeling software using the generic block library and controls toolbox feature.

Appendix C and F are the MATLAB program and SIMULINK model developed by Leonard. Leonard's work is labeled Program L and Model L. Appendix B and F contain the program, Program H, and model, Model H, developed by the author.

Two of the three thesis objectives were to determine the validity of Leonard's design process through an independent drafting of the ACS program and model. This objective was met. Figures 3.8 and 3.9 display the contour plots for each magnetic model of the Earth. As

previously discussed in Chapter II, the results were compared to a model developed by the Finnish Meteorological Institute. Moreover, the models compared well to illustrations in [1]. Constants, gains, moments of inertias, initialization values, and variable declarations were kept the same in Program H as declared in Program L. This was accomplished to maintain continuity because of the complex nature of the computer code. However, there were differences in the drafting of the program.

Program H's magnetic field model is taken to the 10^{th} degree and 10^{th} order, vice Leonard's calculation of ephemeris data using 8^{th}-order values. Moreover, Program H corrects the IGRF data to 2005 whereas Program L does not. Upon launch, the ephemeris data loaded into the spacecraft's computer will require 2005-2010 IGRF calculations. Furthermore, as NPSAT1 ages, the ACS program will require periodic updates to keep the ephemeris data current and corrected.

IGRF 2000–2005 data used in Program H was downloaded from a database owned and updated by the National Oceanic and Atmospheric Administration (NOAA). [10] The database is managed by the NOAA National Geophysical Data Center (NGDC).

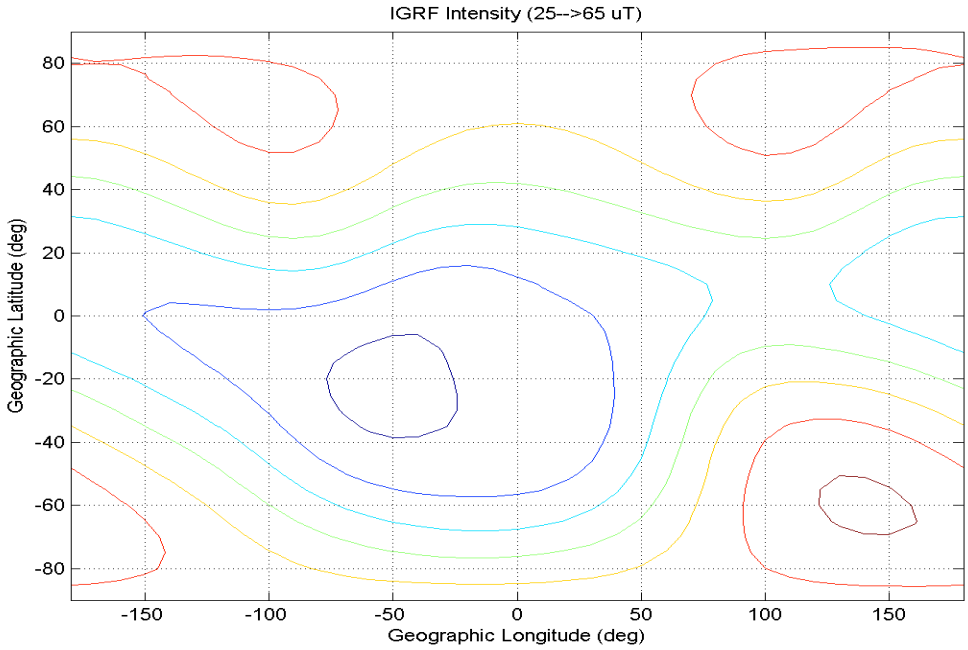


Figure 3.8 Program L IGRF 2000 Uncorrected Earth Magnetic Field Contour Plot (8th order, 8th degree) at 600 km Above the Earth [From Ref. 9.]

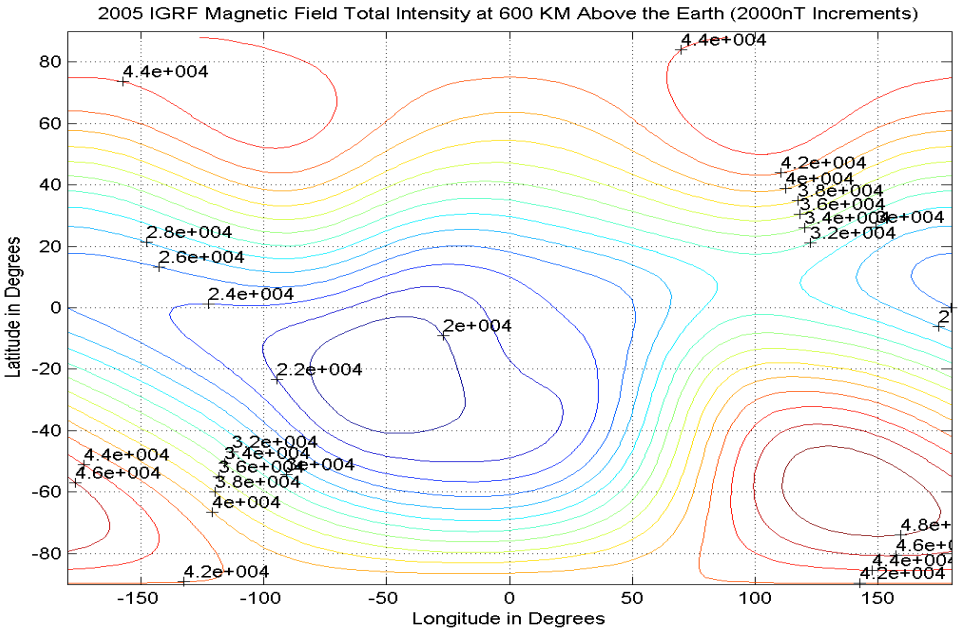


Figure 3.9 Program H IGRF 2000 Corrected Earth Magnetic Field Contour Plot (10th order, 10th degree) at 600 km Above the Earth

The second difference is regarding the spacecraft MOI values. For test and simulation purposes, the MOI values used in Program H were those identified by Leonard. However, as the satillite is manufactured and assembled, the moments of inertia will change and will require updating. Furthermore, once the MOI values are updated, additional computer and air–bearing simulations will be required to enhance the NPSAT1 team's confidence in the ACS. Since NPSAT1 is symmetric and designed to be gravity–gradient friendly, simulation results should not change to the point of instability; however, good engineering practice dictates conservativeness and rigorous testing.

NPSAT1 ACS Model L and H are essentially identical. Both models are presented in Appendices E and F, respectively. Model L is Leonard's original construction, while Appendix E shows the section–by–section representation. Model H was constructed based upon the mathematical theory discussed in Chapter III, Sections B–F. One difference between Model L and H is that each model uses the notations developed in their respective MATLAB M-files. The SIMULINK model validation was completed based upon the empirical relationships discussed in Chapter III. Each model's sub-sections were built separately and compared to Model L to ensure Leonard was following the control theory he cited. No problems were

discovered in the inspection. All of Leonard's model relationships corresponded with ACS control theory and application to that presented in [6].

In summary, this chapter identified and explained all of the empirical relationships used to design the NPSAT1 attitude control system. The relationships discussed were used to build a model used to mimic the NPSAT1 attitude positioning. Furthermore, once the mathematical relationships were established, the models and programs developed in this thesis were compared against Leonard's work. No substantial issues were found. Specifically, the MATLAB program used as to determine the ephemeris magnetic field values and ACS model entering arguments computed correctly. Additionally, the empirical inspection of Leonard's model found that his structure was sufficient and followed ACS control theory. The next step was to transition from a computer modeling simulation phase to a hardware testing phase. However, to do so required the establishment of a test platform and laboratory to conduct hardware tests. Discussion of the laboratory creation is discussed in Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. NPSAT1 AIR–BEARING LABORATORY ESTABLISHMENT AND MAGNETIC FIELD SURVEYS

A. OVERVIEW

This chapter discusses the preliminary steps taken to commence the NPSAT1 ACS *hardware-in-the-loop* testing that was designed to demonstrate the effectiveness of the ACS program. However, to reach this *hardware-in-the-loop* testing objective, a test–bed was needed. Therefore, the establishment of an experimental laboratory designed to use an air–bearing platform to prove the ACS program's validity was required.

The creation of the test setting required three goals to be accomplished. These were to find a suitable laboratory that would 1) provide an environment without circulating air, 2) possess a foundation level that was strong enough to affix the air–bearing platform to, and 3) prove to have a measureable static magnetic field. Based on the aforementioned criteria, the air–bearing platform was placed in the SSAG Satellite Clean Room located in Bullard Hall at NPS.

The laboratory floor was covered in electrically–safe rubber matting. The matting provided a good surface to draw a reference grid upon as well as providing a suitable covering for the air–bearing platform to be set upon.

Section B of this chapter discusses in more detail how the laboratory's grid reference system was created and how its magnetic signature was recorded.

B. LABORATORY ESTABLISHMENT

The NPSAT1 air–bearing laboratory was established based upon preliminary research performed by Schmidt. [11] Specifically, Schmidt built an air–bearing test platform to simulate the NPSAT1 ACS. The computer–generated depiction of Schmidt's air–bearing is shown in Figure 4.1. The test platform installed in the new laboratory is shown in Figures 4.2 and 4.3. However, Schmidt did not perform any "hardware-in-the-loop" tests.

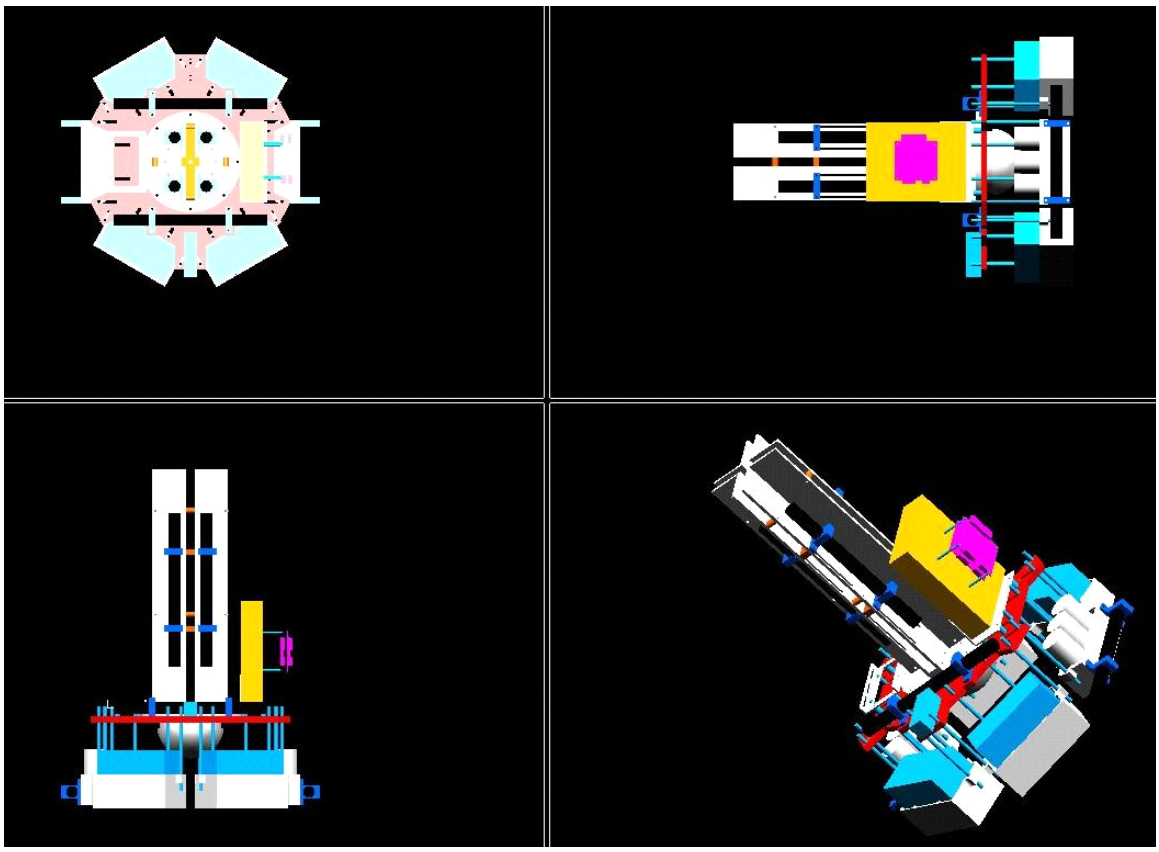


Figure 4.1 NPSAT1 Air–Bearing Test Platform Computer–Generated View [Provided by Mr. Dan Sakoda, NPS SSAG]

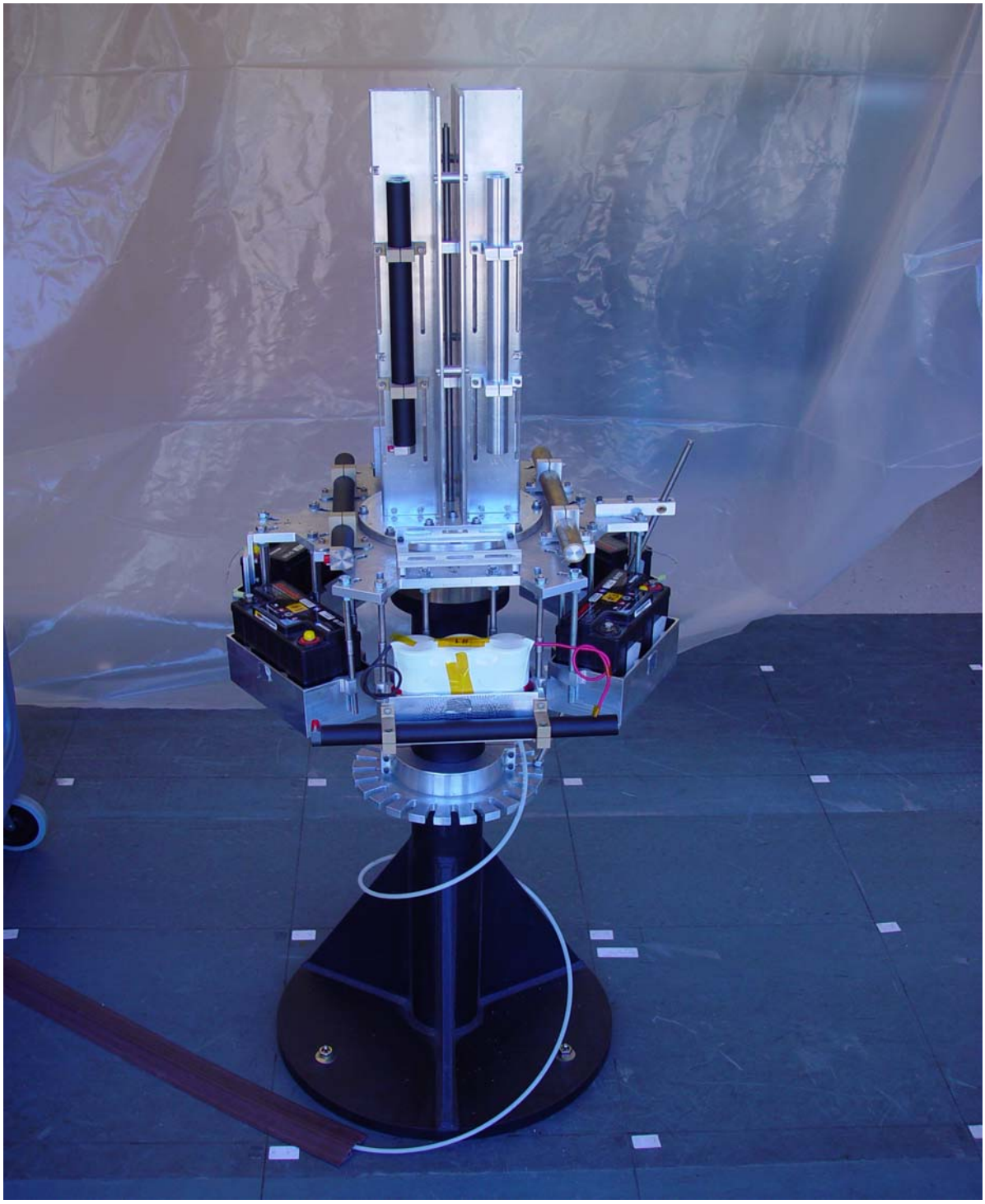


Figure 4.2 NPSAT1 Air–Bearing Test Platforms with *X,Y,Z* Torque Rods, Ballast, and Batteries Installed

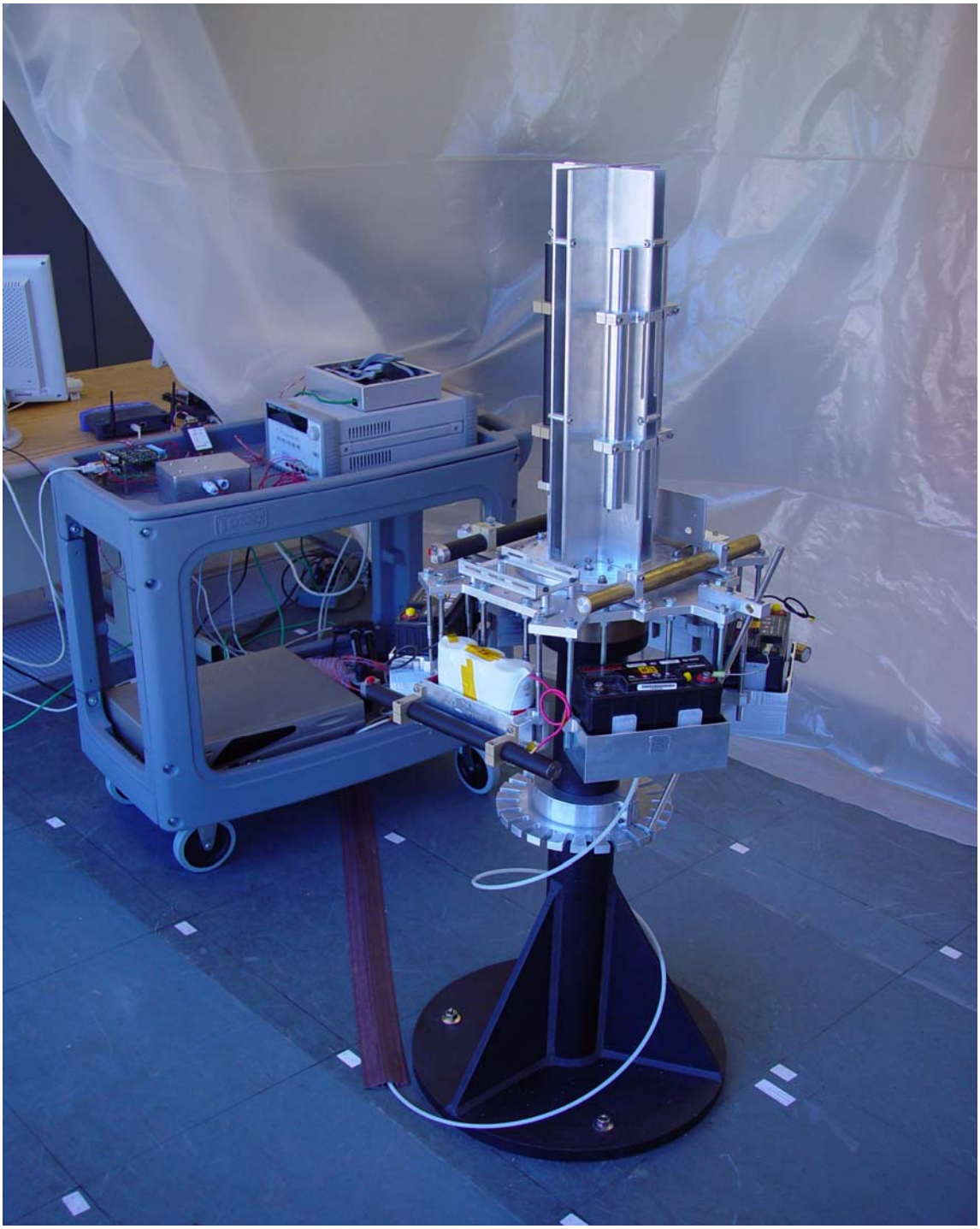


Figure 4.3 NPSAT1 Air–Bearing Test Platforms with CPU, Torque Rod Control Board, Magnetometer, and Power Distribution Box Set for Preliminary Tests

Reference [11] contained a preliminary magnetic signature survey of the laboratory where the air–bearing was to be housed. However, building renovations following completion of the survey and prior to commencement of this thesis forced the movement of the air–bearing platform to a new location. Therefore, new and more robust laboratory magnetic surveys were completed using magnetometers and MATLAB programs vice the MAPLE software program used in [11].

The air–bearing platform and associated hardware are pictured in Figures 4.2 and 4.3. The magnetic surveys commenced with the establishment of a grid reference system. The grid reference was created in a matrix form, facilitating an efficient method to measure and save magnetometer readings into MATLAB variables.

The reference system was composed of a 5 x 7 grid. Each block of the grid was 0.5 meters long by 0.5 meters wide. The grid is referenced by *i* and *j* index values. These are the corner points of the grid. Starting in the lower left corner, the *i* index values start at one and increase to eight. The *j* index values commence at one and increase to six. Additionally, the grid's area above the floor was considered. Four planes were selected and noted as the *k* index. The levels measured were 0 mm, 977 mm, 1250 mm, and 1517 mm above the floor. The floor was considered to be located at sea level. The *i, j, k* matrix was a rectangular Cartesian grid. Figure 4.4 represents the grid's orientation. The center circle identifies the air–bearing platform's location within the grid.

Three discrete magnetic surveys were completed using a Honeywell HMR2300 magnetometer. "The Honeywell HMR2300 is a three-axis smart digital magnetometer [designed] to detect the strength and direction of an incident magnetic field. The three… Honeywell magneto-resistive sensors are oriented in orthogonal directions to measure the *X*, *Y*, and *Z* vector components of a magnetic field." [12, p. 1] Figure 4.5 displays the magnetometer used during the surveys.

Data was collected using a DOS program supplied by Honeywell. Magnetometer specifications were set using this program. Table 4.1 lists the settings used during the surveys and as described in [12]. These settings were used throughout the air-bearing tests.

The magnetometer was secured to a non-ferrous tripod centered at each corner of the reference grid. Measurements were taken at discrete points and recorded for 10 to 15 seconds.

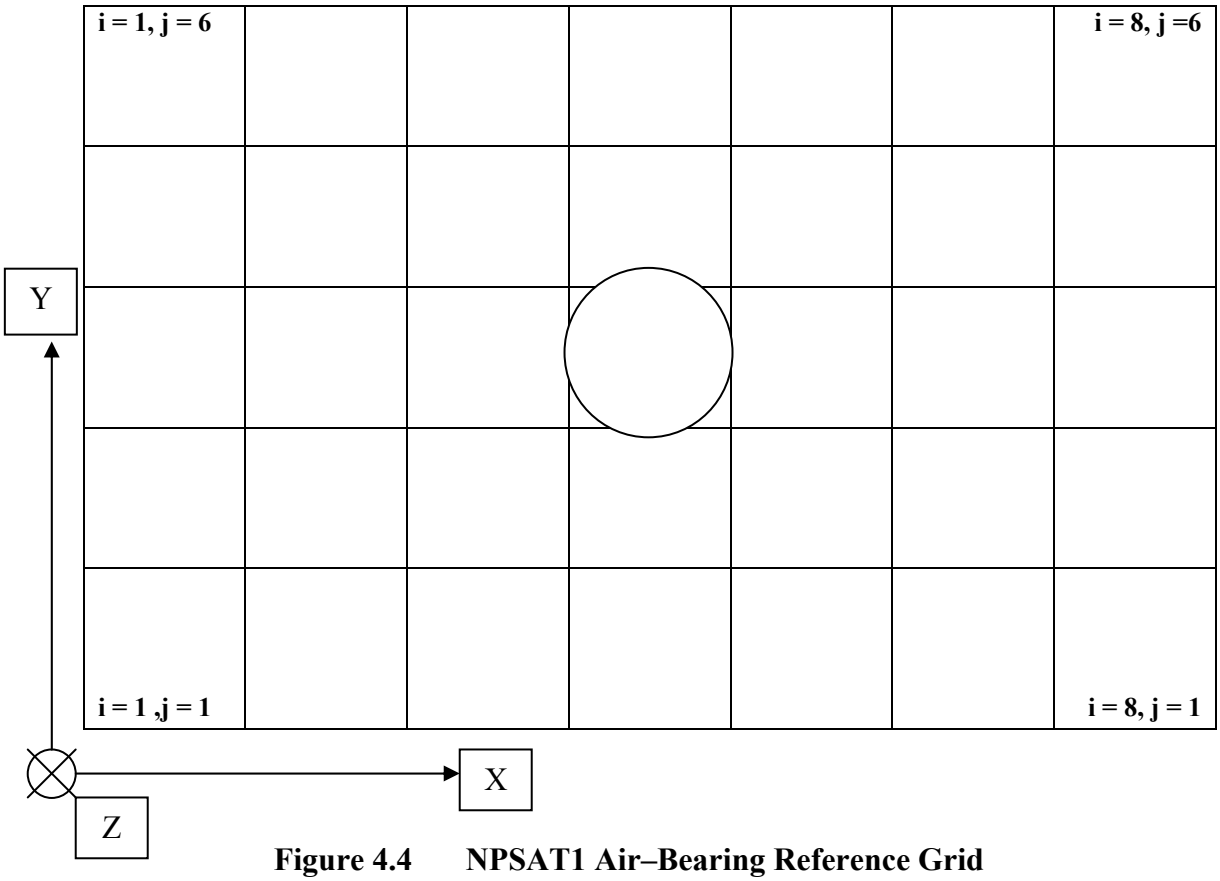


Figure 4.4 NPSAT1 Air–Bearing Reference Grid

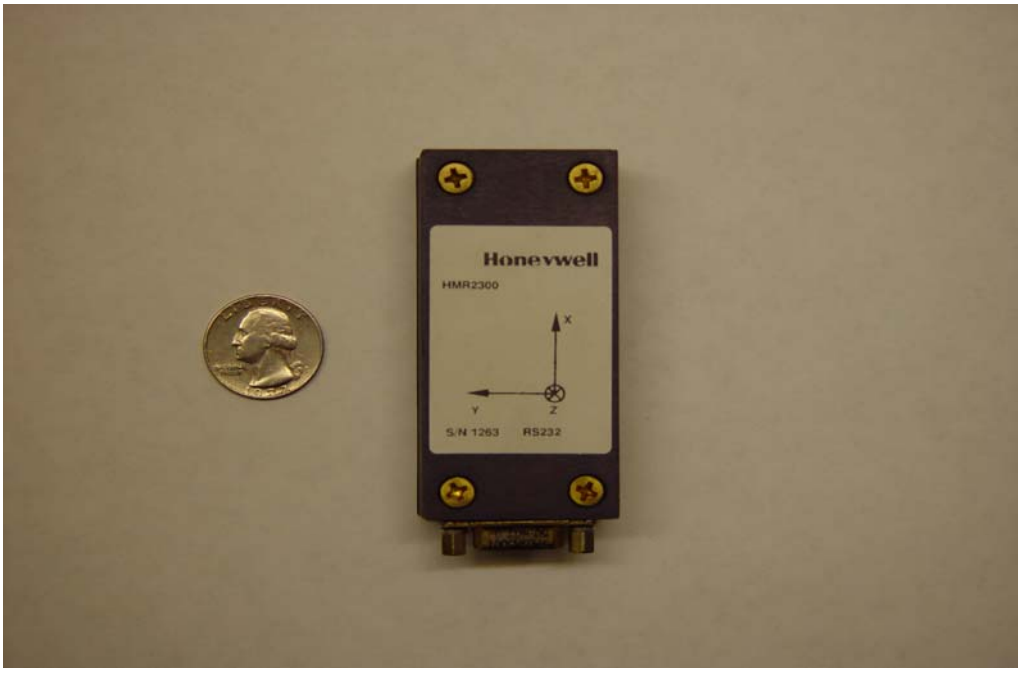


Figure 4.5 Honeywell HMR2300 3-Axis Magnetometer

Average Readings	Baud Rate	Format	ID	Output	Plot Scale	Re-enter	Sample Rate	Set / Reset	Zero Reading
Off	9600 bps	ASCII	01	Polled	± 600 mGauss	Off	20 sps	On	Off

Table 4.1 Honeywell HMR2300 Magnetometer Settings

The magnetometer three-axis orientation was different than the laboratory reference grid. Therefore, conversion from the magnetometer axes to Cartesian axes was accomplished using the following relationship

$$\begin{bmatrix} X_C & Y_C & Z_C \end{bmatrix}^T = \begin{bmatrix} Y_M & -Z_M & -X_M \end{bmatrix}^T . \tag{4.1}$$

The subscripts *C* and *M* represent the Cartesian and magnetometer axes, respectively.

The HMR2300 magnetometer displays four readings. The first three are the *X*, *Y,* and *Z* field components. The fourth element is magnetic North. The latter term was not needed; therefore, it was disregarded and extracted from field measurements.

The magnetic field components are measured in counts. Each measurement can be converted to either Gauss or to Tesla units. One Gauss unit equals 15,000 counts. Conversion to Tesla was completed using

$$\left(X \text{ counts}\right)\left(\frac{1 \text{ Gauss}}{15,000 \text{ Counts}}\right)\left(\frac{1 \text{ Tesla}}{10,000 \text{ Gauss}}\right) = \left(X\right)\left(1.5\times10^{-8} \text{ Tesla}\right) \tag{4.2}$$

The ACS IGRF ephemeris data was calculated in Tesla units; therefore, the air–bearing laboratory magnetic field calculations used the same units.

The air–bearing magnetometer was compared against a calibrated HMR2300 unit prior to measurement to ensure for correct readings. Table 4.2 lists the differences between the magnetometer used for magnetic field measurement and the calibrated unit. Table 4.2 values are displayed in Counts.

	X	Y	Z
Field Magnetometer	0.1315	0.2189	–0.3682
Calibration Magnetometer	0.1740	0.1987	–0.3718
Delta	0.0425	–0.0202	–0.0036

Table 4.2 Honeywell HMR2300 Calibration (Counts)

Each of the three fields yielded a contour and gradient vector field representation. Figures 4.6–4.17 show the magnetic field for the entire air–bearing reference grid. The *X* and *Y* axes are measured in 0.5 meter increments while the *Z* axis is measured in Tesla.

Figures 4.6–4.8 are representative of the manner in which all of the measurements were displayed. Colored contour plots allows ones to visualize how the magnetic field changes in a finite area. Focus should be applied to the points where the contour plot's grid points intersect. The intersection points correspond with the discrete points measured within the grid reference system. The changing colors represent the gradient field changes. Figure 4.7 is a characterization that should be used with Figure 4.6. Instead of attempting to read the magnetic field's magnitude on the vertical axis of Figure 4.6, interpretation of the field's strength is more readily available from Figure 4.7. Additionally, Figure 4.8 was designed to display the manner in which the magnetic field travels. This type of representation was used in all three magnetic field surveys.

The second and third surveys were completed based on the results received from the first survey. As exhibited by Figures 4.6–4.17, the magnetic field was warped. This was an expected condition due to the large amounts of metal surrounding the laboratory and air-bearing test station. Specifically, the laboratory floor, walls, and ceiling contain metal re–bar rods used during building construction. Additionally, to facilitate laboratory wiring, channels are set in the floor that were covered by diamond–patterned metal plating. Lastly, toolboxes, metal support beams, and NPSAT1 test equipment were located in the laboratory. Though it would be desirable to have all of this material removed, it was deemed impractical.

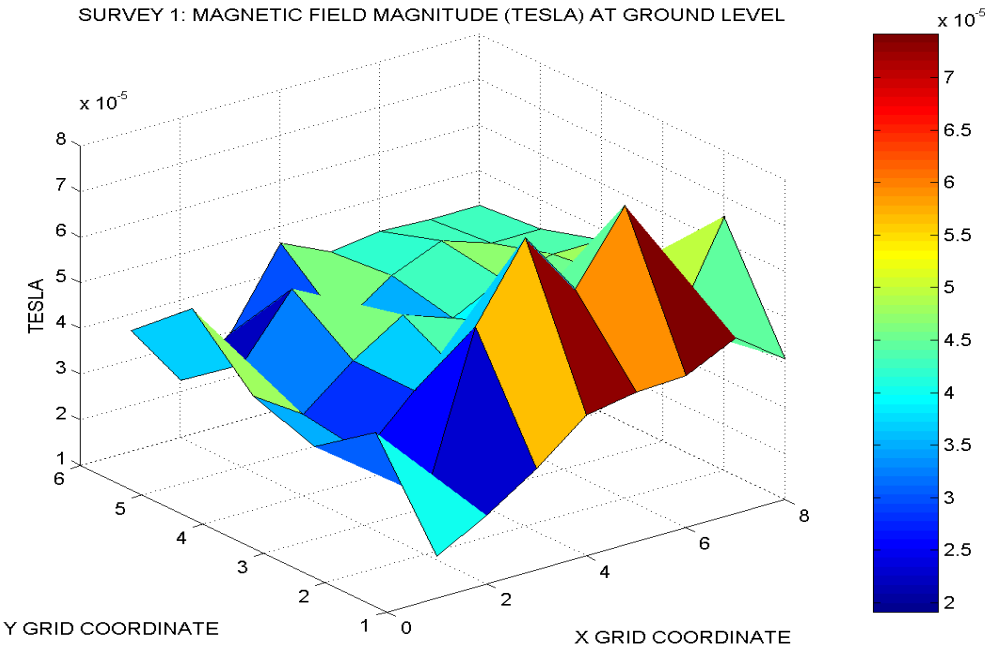


Figure 4.6 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at Floor Level

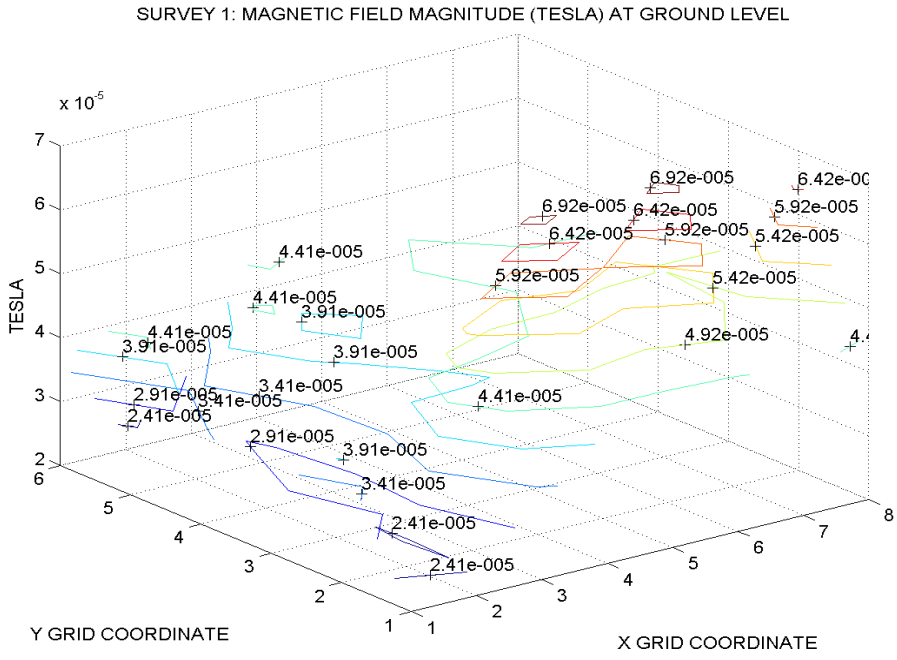


Figure 4.7 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at Floor Level

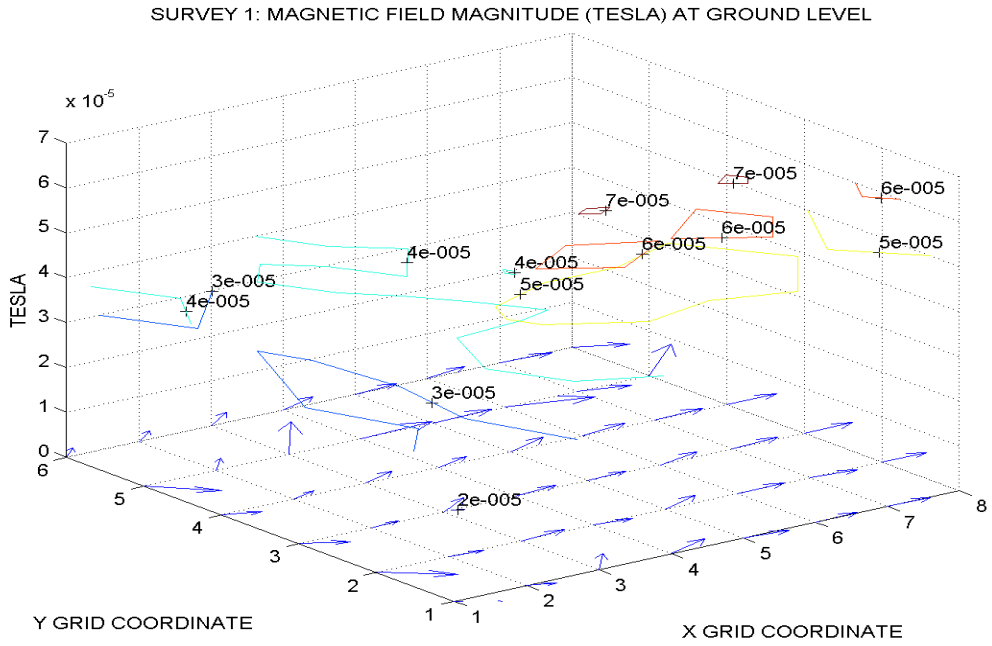


Figure 4.8 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Vector Plot at Floor Level

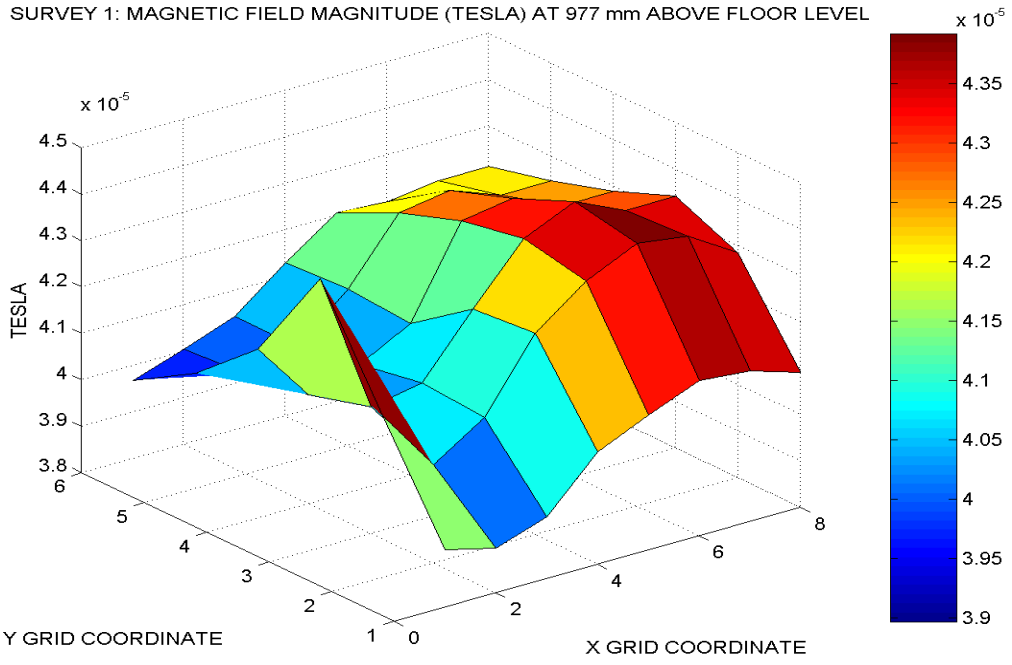


Figure 4.9 NPSAT1 AirBearing Laboratory Magnetic Field Survey 1 Contour Plot at 977 mm Above Floor Level

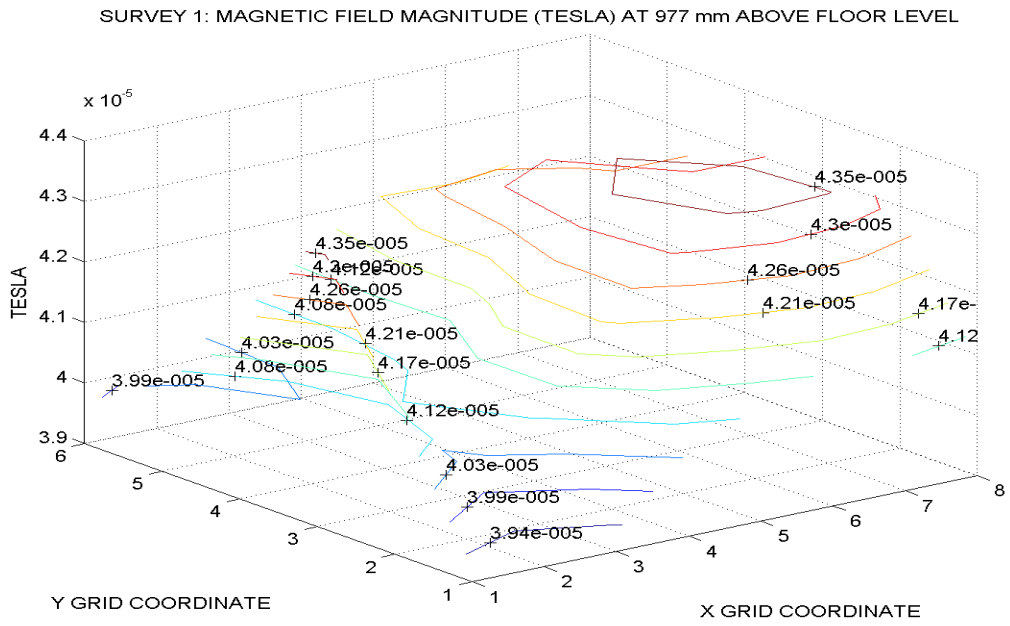


Figure 4.10 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at 977 mm Above Floor Level

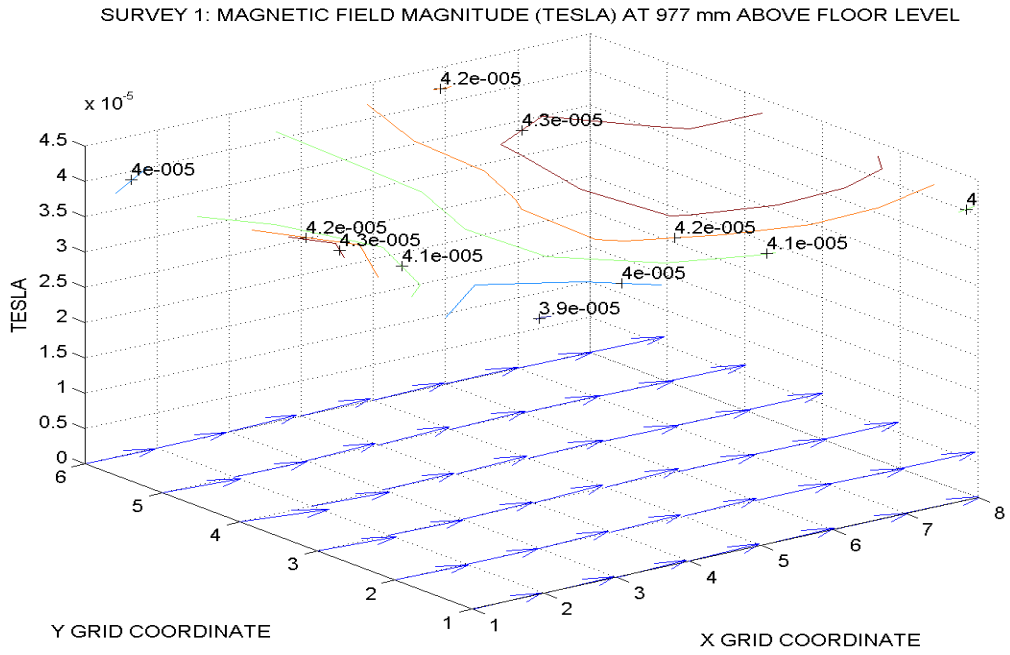


Figure 4.11 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Vector Plot at 977 mm Above Floor Level

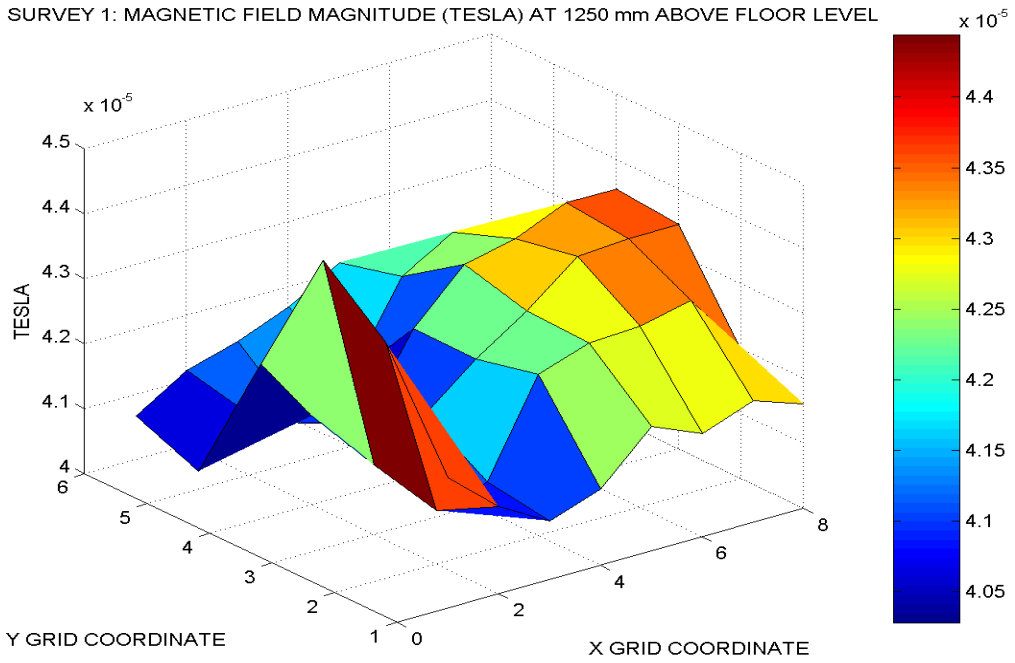


Figure 4.12 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at 1250 mm Above Floor Level

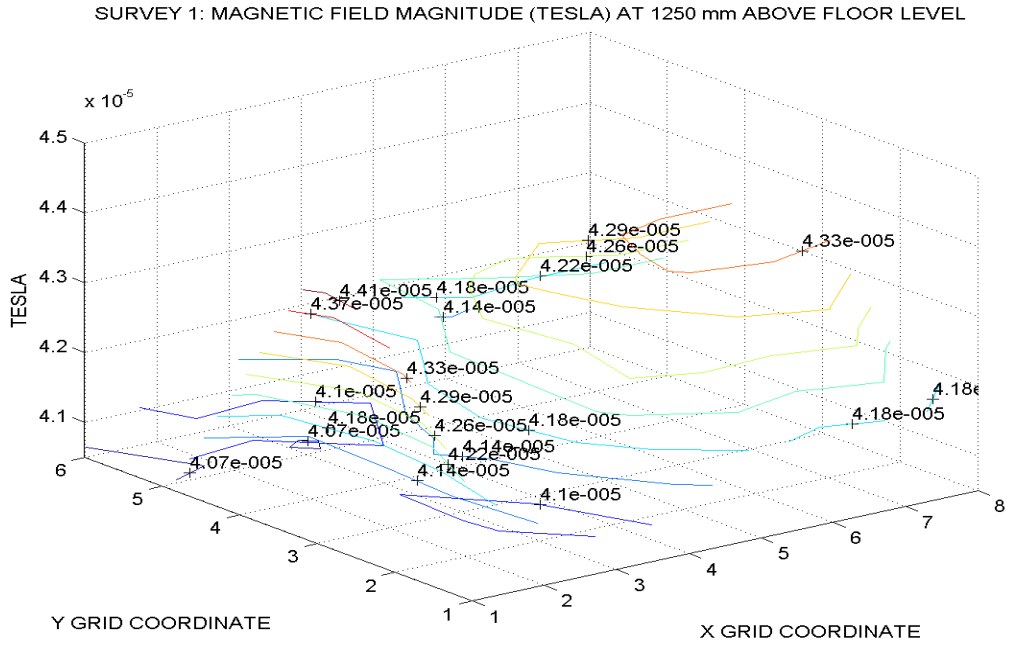


Figure 4.13 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at 1250 mm Above Floor Level

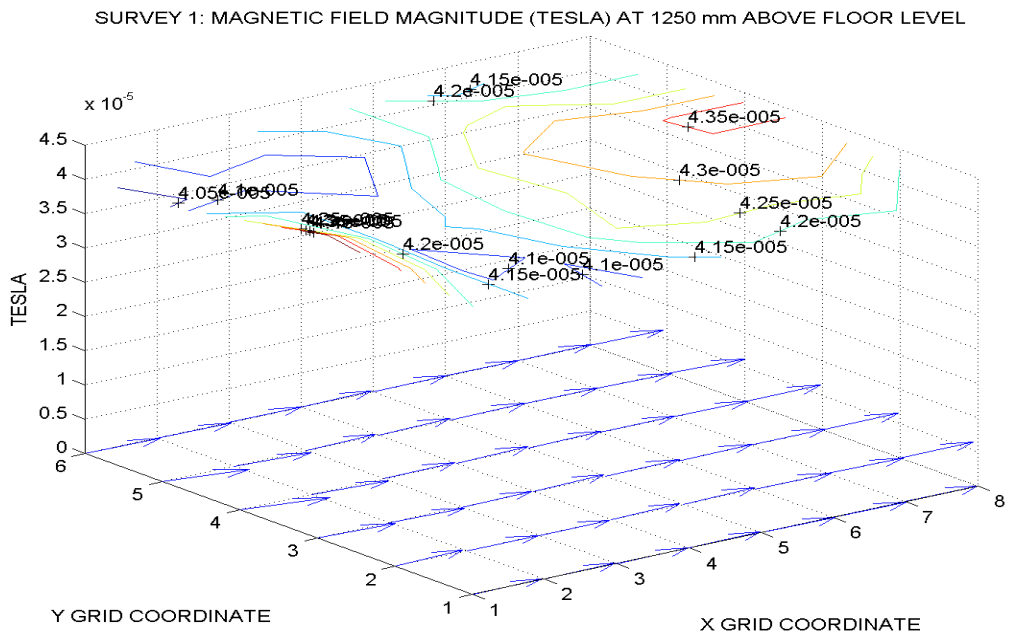


Figure 4.14 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Vector Plot at 1250 mm Above Floor Level

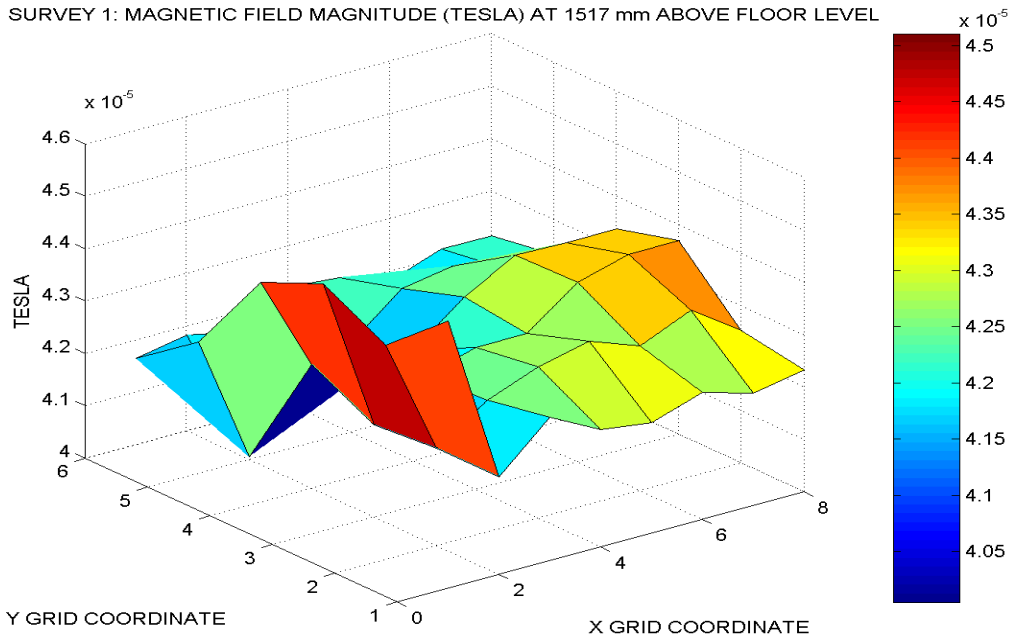


Figure 4.15 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at 1517 mm Above Floor Level

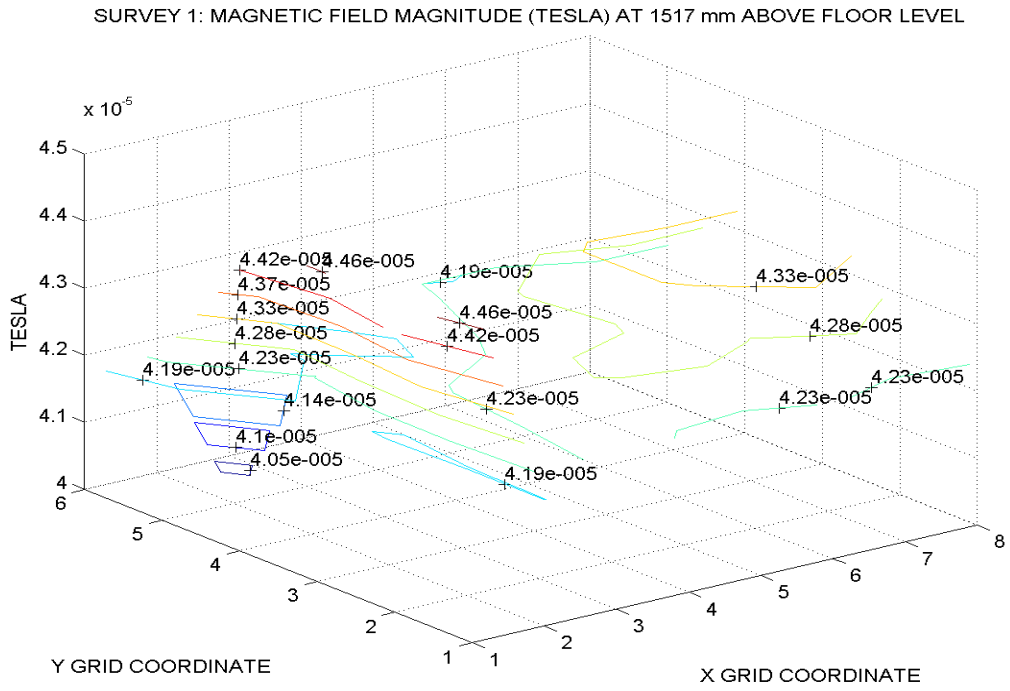


Figure 4.16 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Contour Plot at 1577 mm above Floor Level

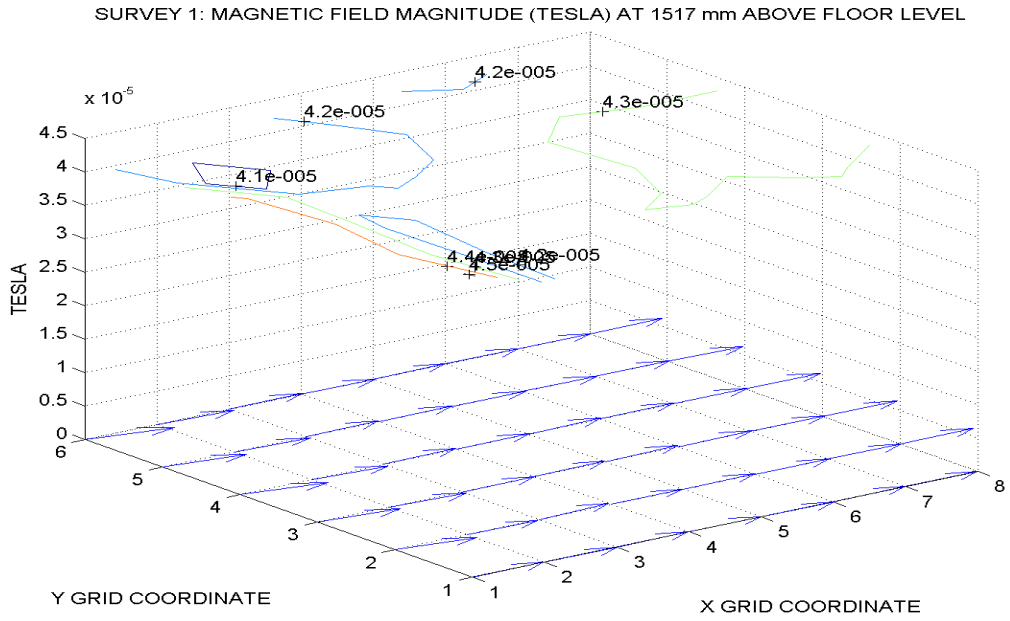


Figure 4.17 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 1 Vector Plot at 1517 mm Above Floor Level

The magnetic field surveys were designed to understand what type of magnetic field resides within the laboratory. The initial survey proved that the field was warped and more refined measurements were required. This was shown by the severe change in the magnetic field at the *Y* equals one to three levels. The large change in field strength corresponds with the location of metal grating on the laboratory's floor. Thus, the second and third magnetic field signature surveys were initiated.

Figures 4.18–4.26 are the results from the second survey. The second survey reduced the grid measurements from 5×7 to 4×4. Moreover, the second survey disregarded floor–level measurements and concentrated the measurements at the 977 mm, 1250 mm, and 1517 mm *Z* axis levels. Additionally, Figures 4.18–4.26 illustrate that the magnetic field was less disturbed and remained fairly static in the vicinity immediately surrounding the air–bearing.

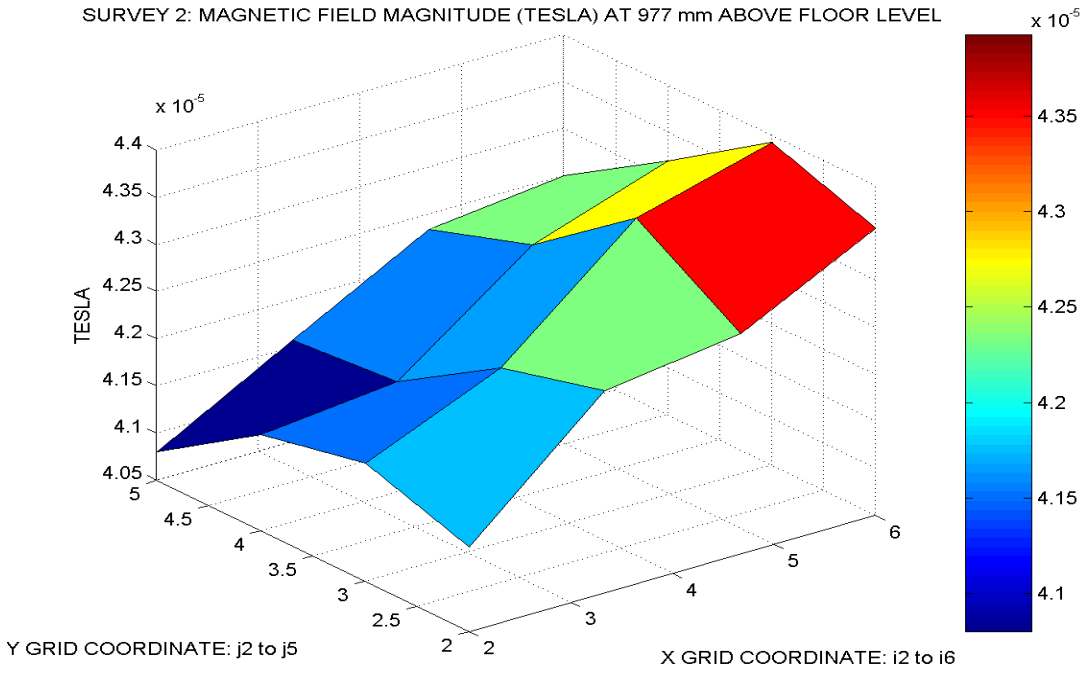


Figure 4.18 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 977 mm Above Floor Level

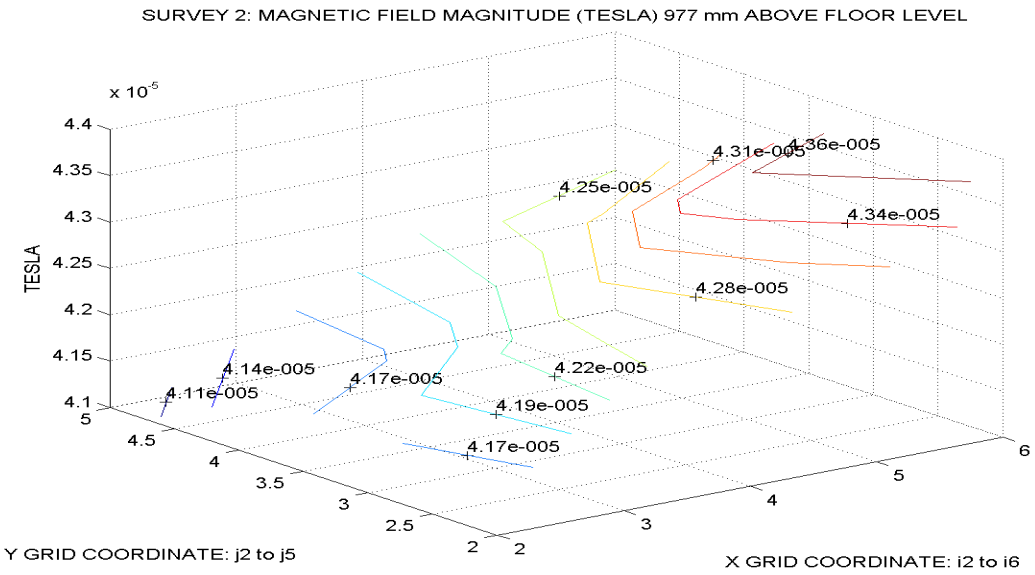


Figure 4.19 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 977 mm Above Floor Level

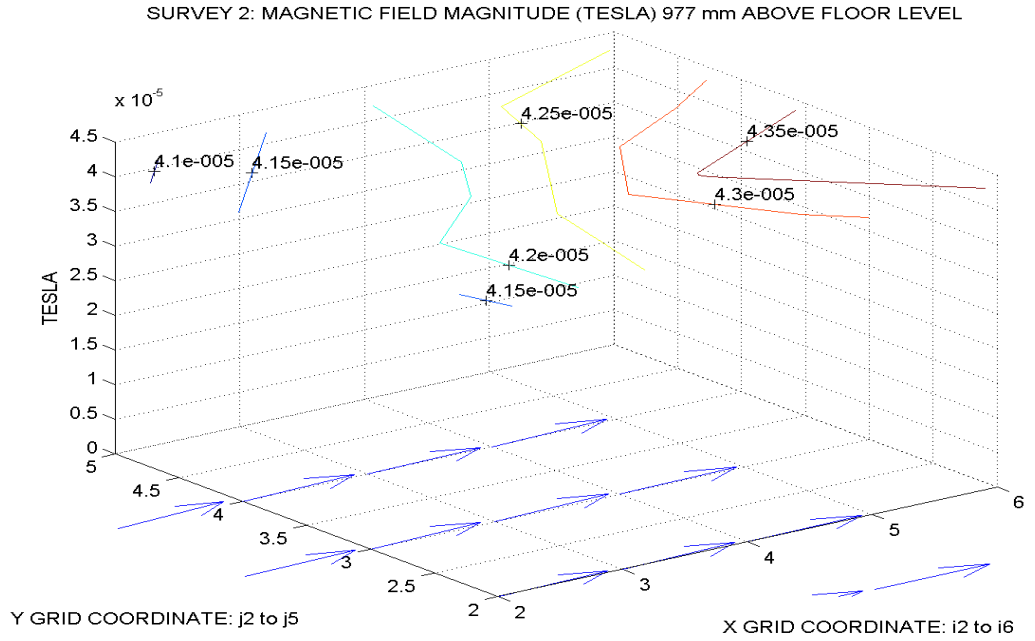


Figure 4.20 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Vector Plot at 977 mm Above Floor Level

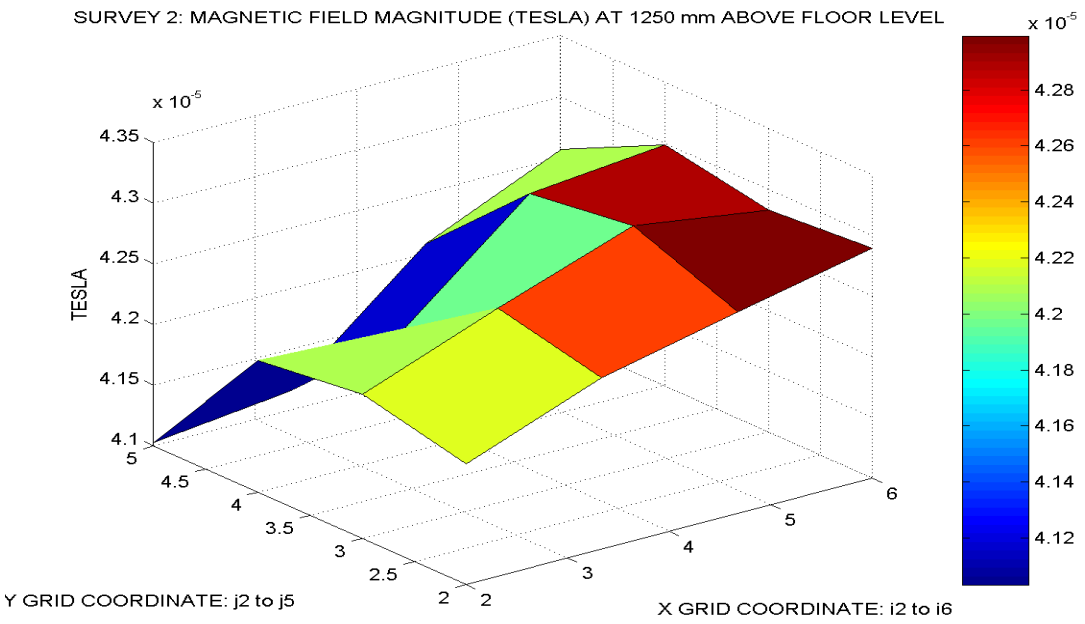


Figure 4.21 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 1250 mm Above Floor Level

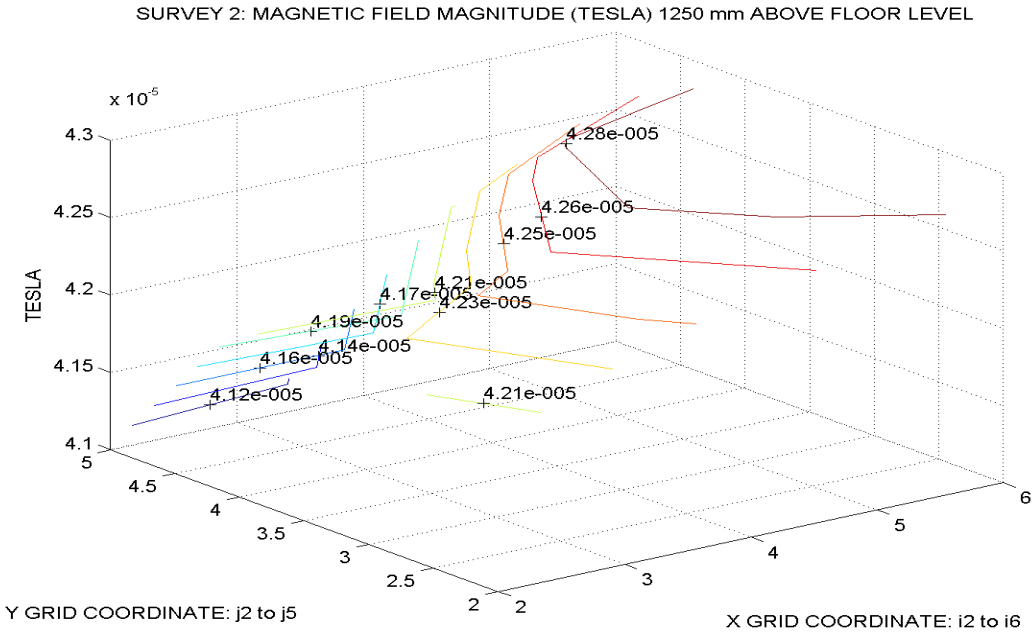


Figure 4.22 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 1250 mm Above Floor Level

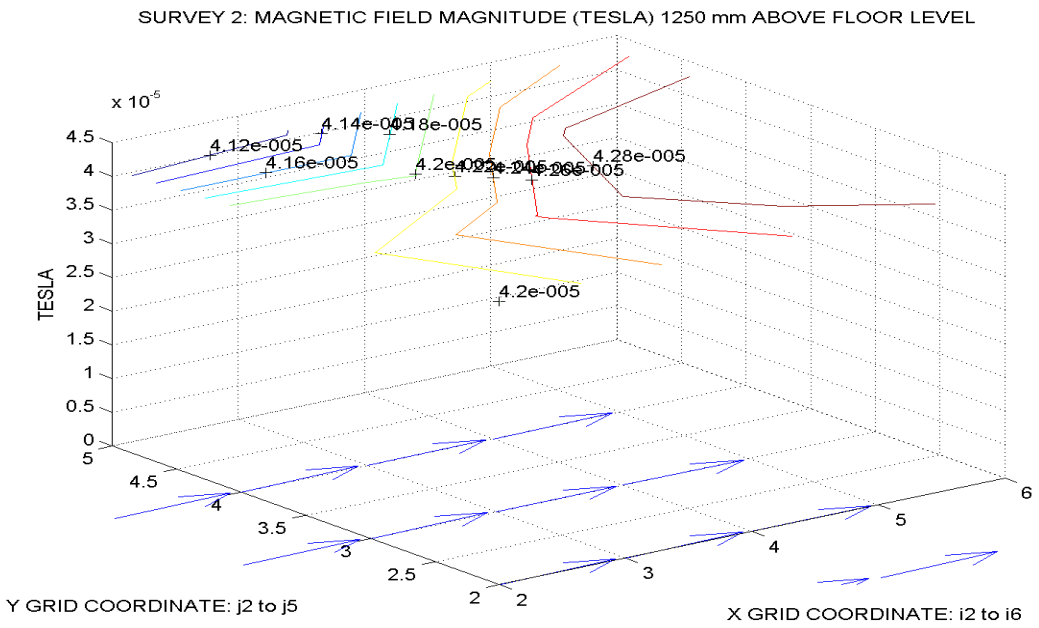


Figure 4.23 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Vector Plot at 1250 mm Above Floor Level

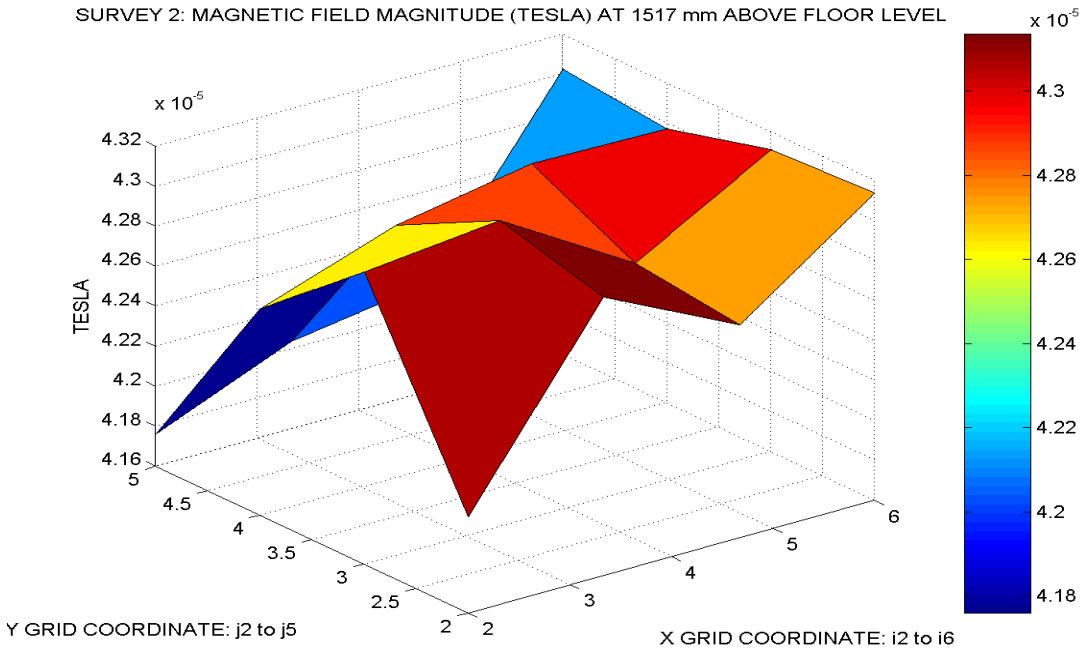


Figure 4.24 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 1517 mm Above Floor Level

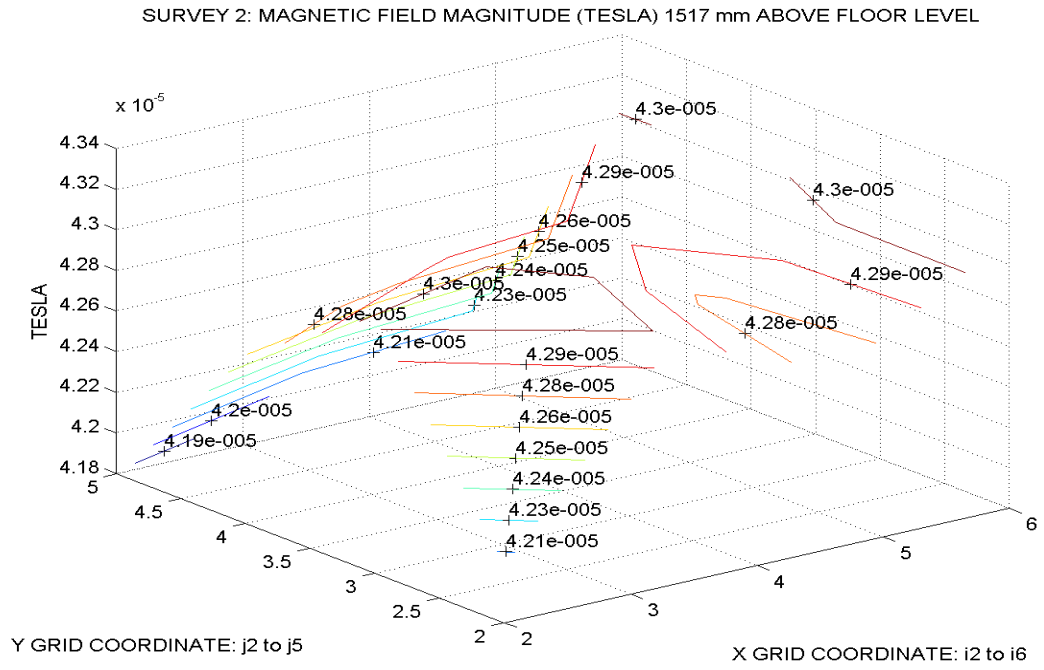


Figure 4.25 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Contour Plot at 1517 mm Above Floor Level

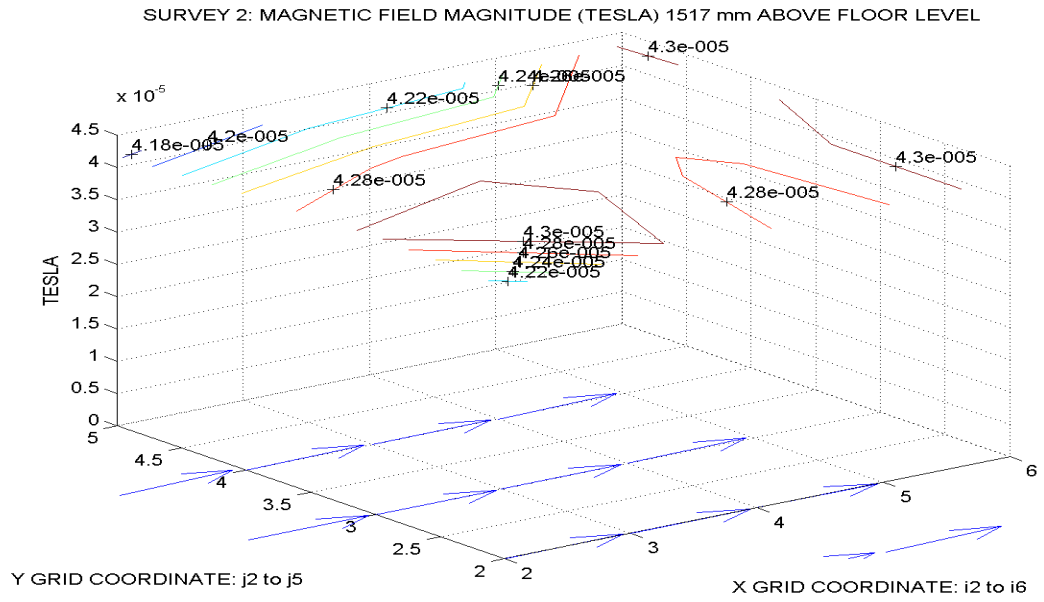


Figure 4.26 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 2 Vector Plot at 1517 mm Above Floor Level

The third air-bearing laboratory magnetic field survey was conducted with the magnetometer fixed to the test platform. Three sets of measurements were recorded. Each set rotated the air-bearing 360 degrees about the *Z* axis. The *Z*-axis rotation angle was noted as the angle ϕ. Additionally, measurements were taken with respect to three inclination angles referenced from the *Z* axis. The angle θ was inclined 75°, 90°, and 105°. These inclination values were labeled Tier 1, Tier 2, and Tier 3, respectively.

Measurements were taken for 20 seconds vice the 10-second increments used during surveys one and two. The MATLAB program, described in Appendix I, used to generate a representative magnetic field model, averaged 280 of the collected data points. Also, the data point's locations were converted from spherical coordinates to rectangular coordinates. The field's direction and magnitude were calculated from the mean values. Figures 4.27–4.35 display the survey's results. Specifically, Figures 4.27–4.29 model the field's direction. Three different aspects are shown to illustrate the field's travel. This coincides with the field direction found in surveys one and two. Figures 4.30–4.35 show each tier's normalized and Tesla field strength. Figure 4.36 combines the magnitudes of all three tiers.

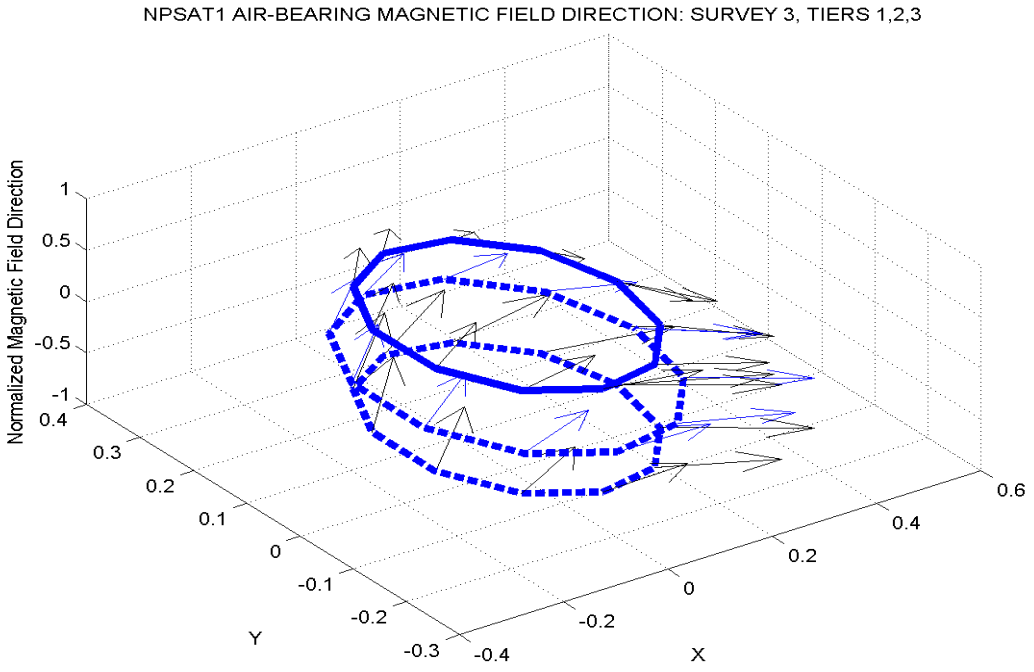


Figure 4.27 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tiers 1, 2, 3 Field Direction: View No. 1

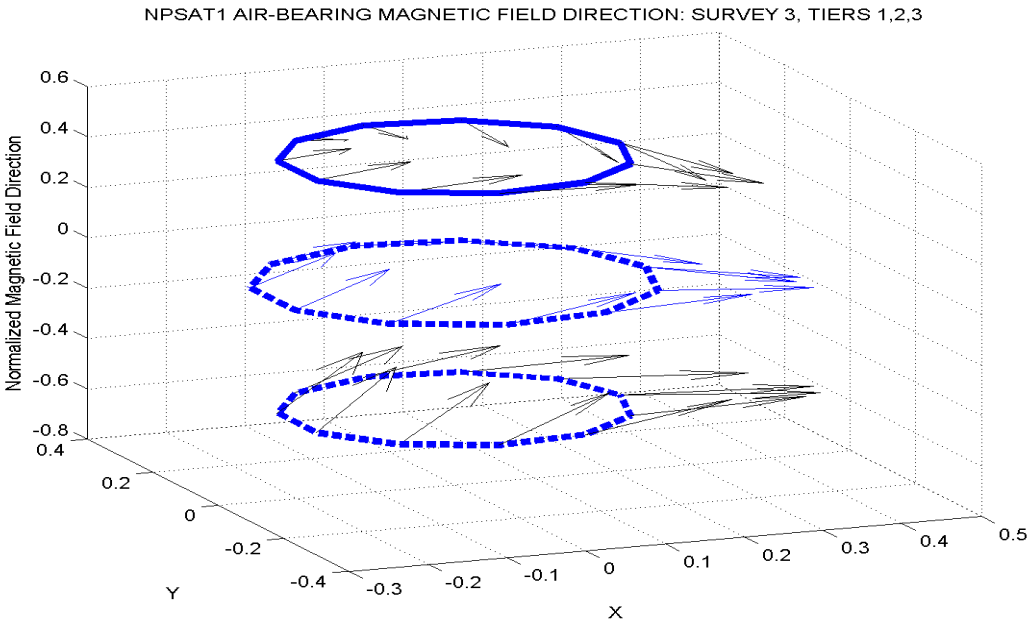


Figure 4.28 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tiers 1, 2, 3 Field Direction: View No. 2

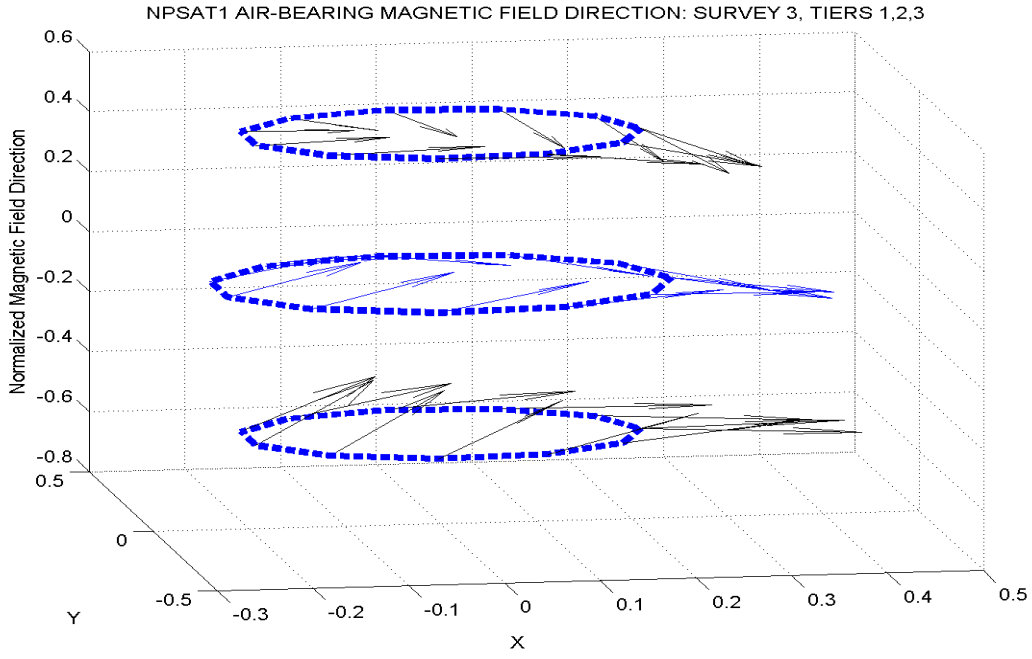


Figure 4.29 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tiers 1, 2, 3 Field Direction: View No. 3

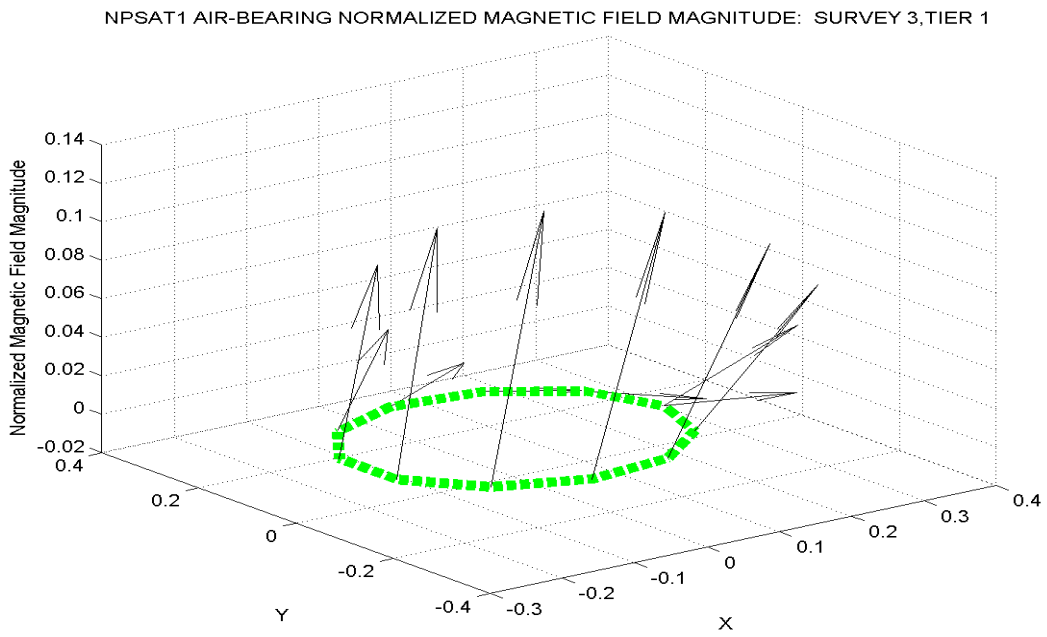


Figure 4.30 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 1 Normalized Field Strength

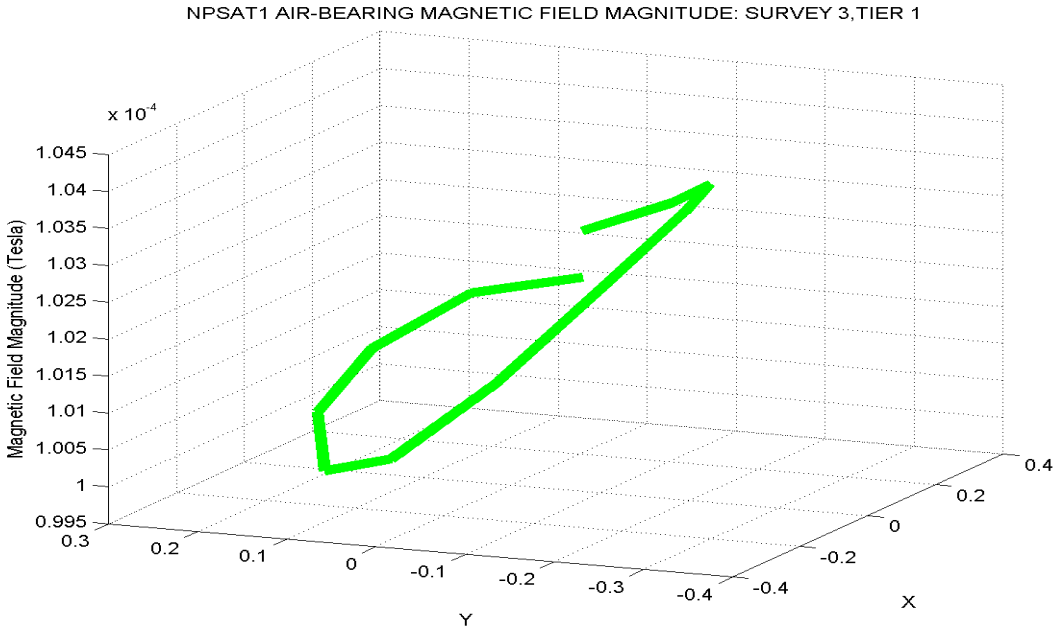


Figure 4.31 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 1 Field Magnitude (Tesla)

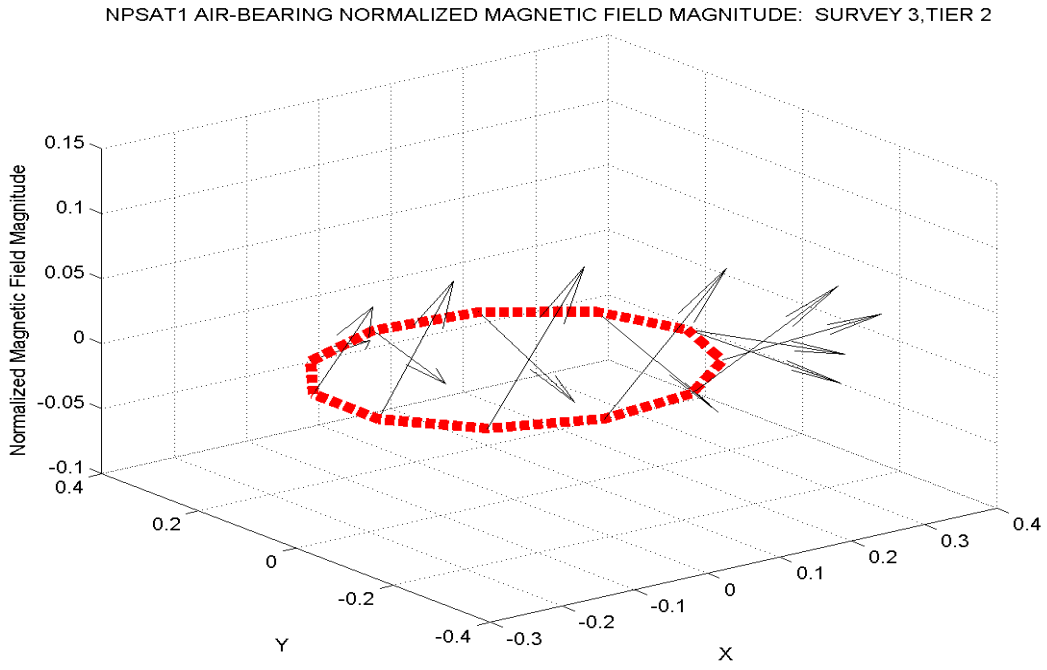


Figure 4.32 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 2 Normalized Field Strength

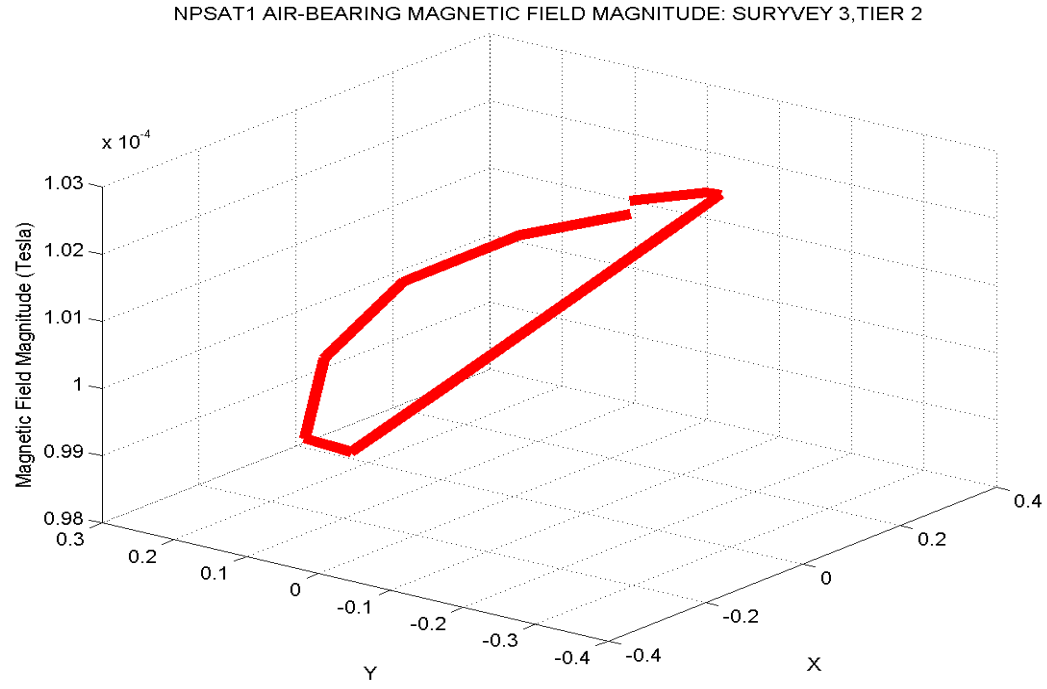


Figure 4.33 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 2 Field Magnitude (Tesla)

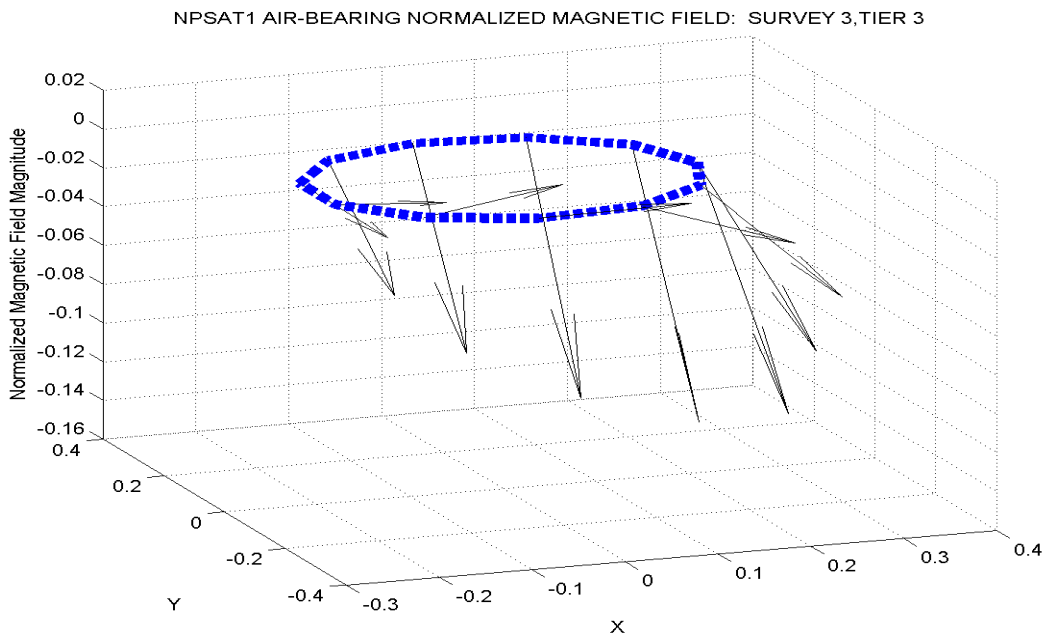


Figure 4.34 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 3 Normalized Field Strength

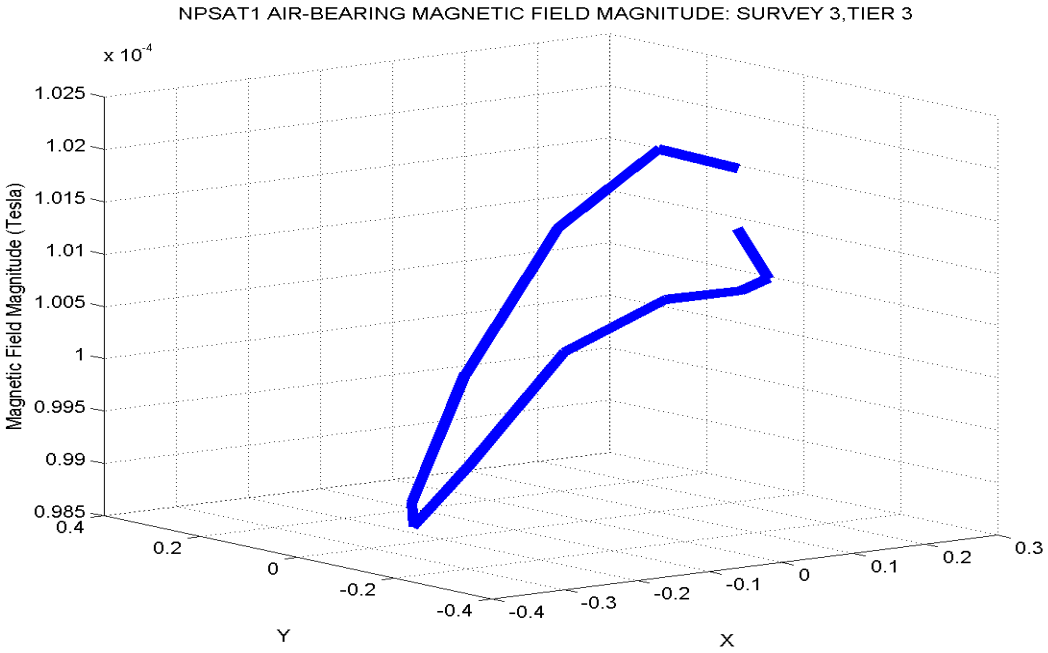


Figure 4.35 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 3 Field Magnitude (Tesla)

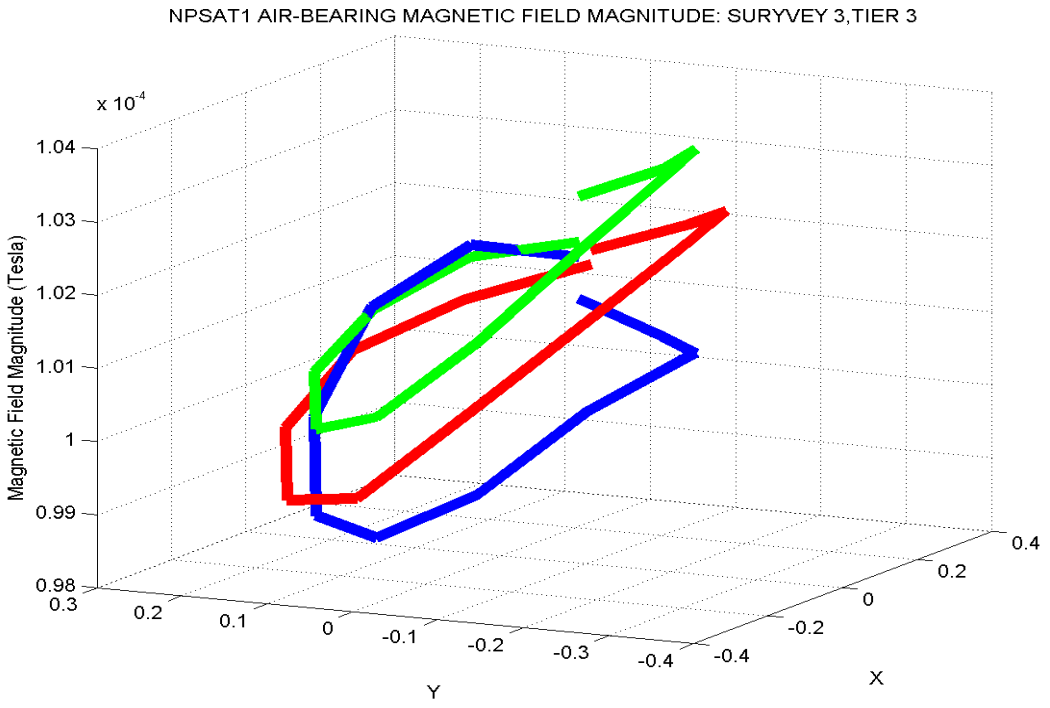


Figure 4.36 NPSAT1 Air–Bearing Laboratory Magnetic Field Survey 3: Tier 3 Field Magnitude (Tesla)

Table 4.3 lists the magnetic field's magnitude (Tesla) at each tier level and sample point. Each sample point represents 30 degrees of the rotation angle θ about the air–bearing platform's *Z* axis. The table was used as the air-bearing's ephemeris data used in the SIMULINK model and ACS air-bearing experiments.

Additional equipment mounted on the air–bearing test platform will include a single board computer, a power supply, a torque driver control board, a wireless remote router, and all associated wiring. Figure 4.37 displays the additional components. Once preliminary experiments are completed, a laser tracking system, cameras, and recording equipment will be installed to allow for a more in–depth analysis of the air-bearing's attitude movement. These have not been installed; therefore, the equipment is not shown.

The single board computer was designed to store the SIMULINK model and support the ACS computing functions. Additionally, the computer was the interface between the ACS and the torque driver board. The torque driver board is a solid–state component designed to control the amount of current applied to the torque rod coils.

Rotation angle, θ , (degrees)	Tier 1 75° inclination	Tier 2 90° inclination	Tier 3 105° inclination
000	1.02E-04	1.02E-04	1.02E-04
030	1.02E-04	1.01E-04	1.02E-04
060	1.01E-04	1.00E-04	1.01E-04
090	9.97E-05	9.96E-05	1.00E-04
120	9.86E-05	9.89E-05	9.98E-05
150	9.86E-05	9.92E-05	1.00E-04
180	9.93E-05	1.01E-04	1.01E-04
210	1.00E-04	1.02E-04	1.03E-04
240	1.01E-04	1.03E-04	1.03E-04
270	1.01E-04	1.03E-04	1.04E-04
300	1.01E-04	1.02E-04	1.03E-04
330	1.01E-04	1.02E-04	1.04E-04

Table 4.3 NPSAT1 Air–Bearing Magnetic Field Magnitude Calculations Survey 3: Tiers 1, 2, 3

The power supply provides all energy needed by the torque rods, computer, wireless router, and torque driver board.

In summary, this chapter identified the methods used to establish a laboratory designed to test the NPSAT1 ACS. Specifically, a reference system was created that allowed for a systematic measurement of the laboratory's magnetic field. The laboratory magnetic field measurements were completed in three stages. Each stage narrowed the scope of investigation which allows one to visualize the field's strength, movement, and stability.

A discussion regarding the tests performed using the air–bearing, its components and the ACS is presented in the next chapter.

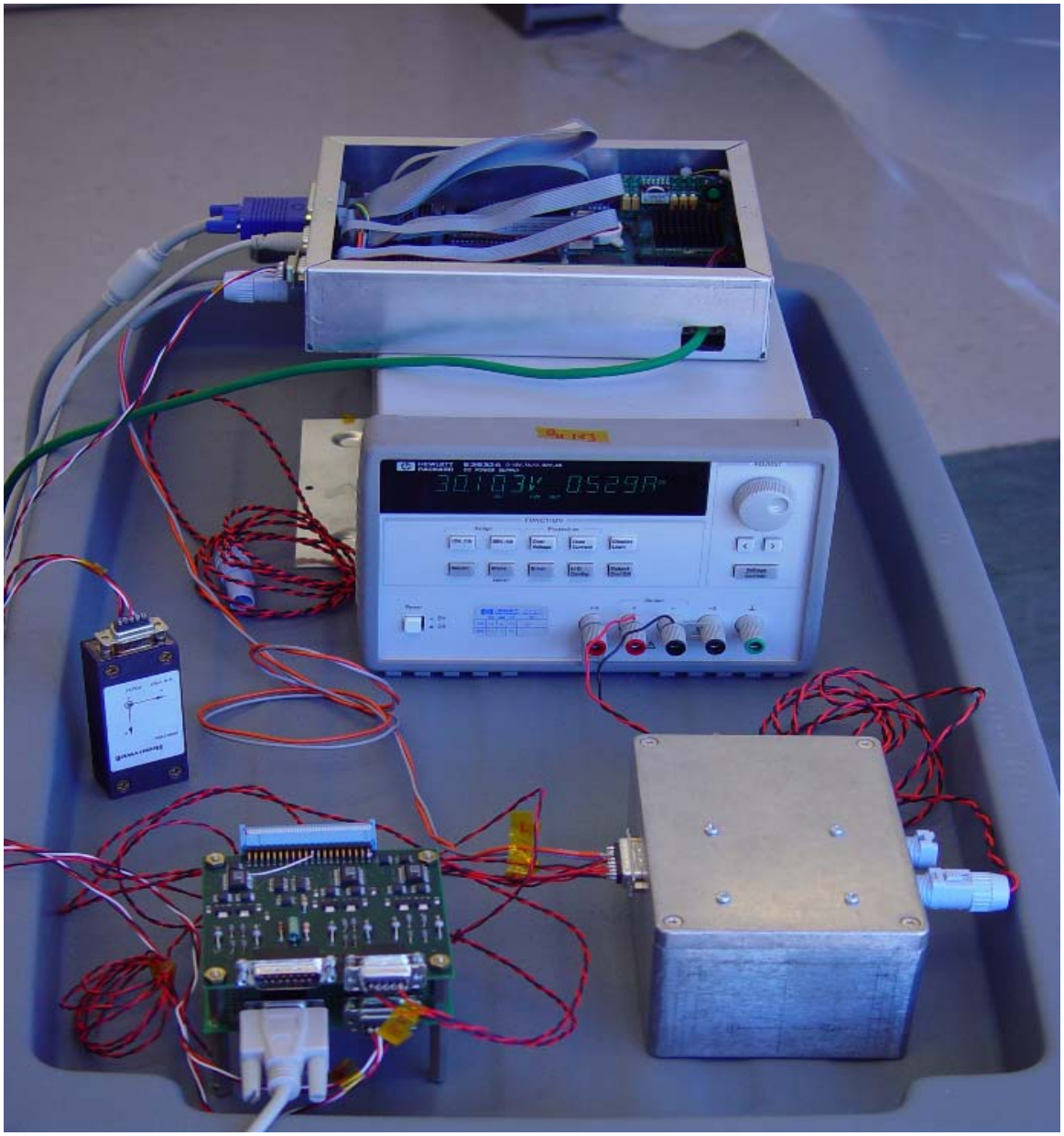


Figure 4.37 NPSAT1 Air–Bearing Test Platform Equipage (Computer, Torque Driver Board, Magnetometer, and Power Supply)

THIS PAGE INTENTIONALLY LEFT BLANK

V. NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM AIR-BEARING TESTS

A. NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM *HARDWARE-IN-THE-LOOP* TEST OVERVIEW

Satisfactory *hardware-in-the-loop* testing of the NPSAT1 ACS program and determination of interfaces function between the computer hardware, power supplies, and magnetic torque rods was the ultimate goal of this thesis. Now, with a laboratory established, *hardware-in-the-loop* were ready to commence. This chapter discusses the tests performed, results found, and solutions initiated during these initial examinations.

B. NPSAT1 MAGNETIC ATTITUDE CONTROL SYSTEM AIR-BEARING TEST PLAN

The major goal of this thesis was to develop and execute tests designed to exercise the NPSAT1 magnetic attitude control system. The test plan was developed to identify hardware and software problems. Testing was conducted in three stages. The tests are presented below with amplification and discussion given in separate sections.

The first test completed was a computer simulation using MATLAB and SIMULINK. Specifically, the reduced order estimator was separated from the overall SIMULINK ACS computer model. The REO estimator was tested using magnetic field values as discussed in Chapter IV.

The second stage was to convert the ACS MATLAB program and SIMULINK model into ANSI 'C' code. This was accomplished using the embedded SIMULINK feature known as Real Time Workshop.

Air–bearing tests commenced once the ACS model was successfully converted to ANSI 'C' code. Preliminary experiments included input/output tests, solenoid current tests, analog-to-digital (A/D) conversion tests, torque rod magnetic field production, and induced dipole moment tests. These were completed using bench support test equipment.

A fourth battery of tests was scheduled; however, these were not completed. The experiments were not completed due to the software and hardware faults discovered during the third battery of air–bearing tests. Specifically, during the input/output tests utilizing Real Time Workshop, XPC Target and XPC Host, magnetometer measurements were not being properly

relayed to the ACS SIMULINK model. Additionally, hardware faults residing in the torque driver circuit board were discovered during the solenoid current tests and analog-to-digital conversion tests. Both the software and hardware design issues had to be corrected for air–bearing tests to resume. These two issues are being examined and corrected by NPSAT1 project engineers.

C. NPSAT1 MAGNETIC ACS PLANT STATIC MODELING

The air–bearing control system consists of the reduced order estimator, cross product steering law, torque control law, *Bdot* control law, magnetic field ephemeris data, and measured external magnetic field values. The air-bearing model does not contain any testing regarding the disturbance torques because the actual space environment cannot be simulated. However, follow–on research to the preliminary experiments includes the installation of Helmholtz Coils. This process will allow research using a magnetic field that can be altered to resemble the space environment.

The first step in the testing process was to model the air-bearing's desired behavior. Ideally, the air–bearing would exhibit a damped oscillation motion as displayed in Figure 5.1. This motion was modeled using the air–bearing simulation model depicted in Figure 5.2. The air–bearing model is separated into five sections. The green sections represent the solenoids and magnetometer. The blue blocks depict the torque–rod driver control board. The yellow blocks represent the SIMULINK ACS model converted to 'ANSI' C code. The orange are blocks used to extract data for analysis. The white blocks are portions of the model used to simulate the dynamics and kinematics of the air–bearing. The model was provided by Leonard for research purposes. Each of the sub–model's development was based upon the mathematical principles discussed in Chapter III. The model's parameters were changed to reflect the air–bearing's characteristics and to use the magnetic field survey data previously collected and discussed in Chapter IV.

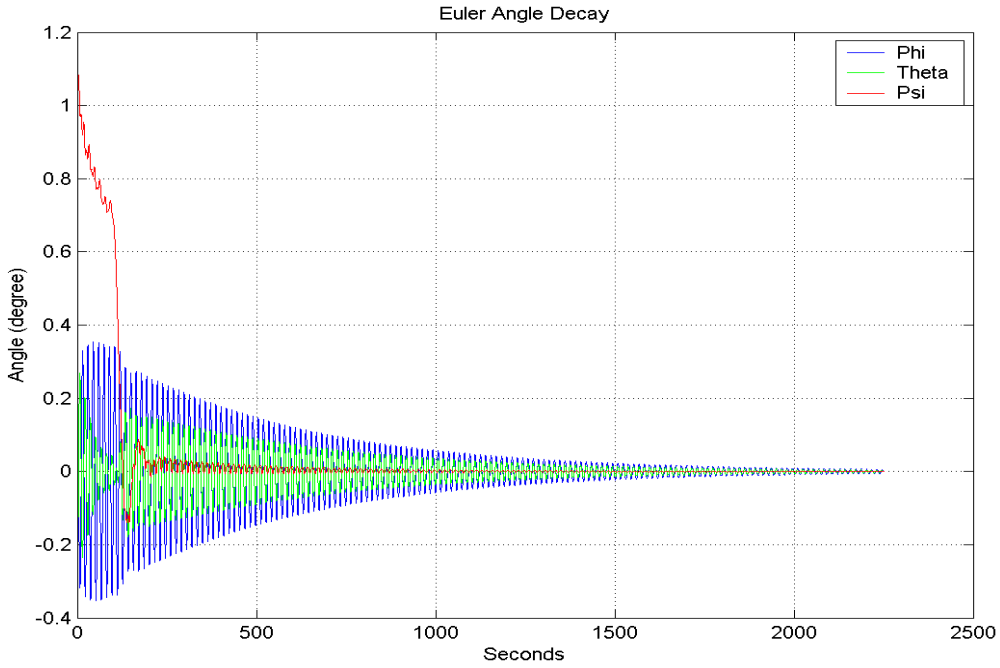


Figure 5.1 NPSAT1 ACS Air–Bearing Simulation Euler Angle Decay

NPSAT1's mission is to remain on orbit at an altitude of 600 ± 40 km. While on orbit, the ACS was designed to have the spacecraft's attitude movements resemble those of a pendulum. Thus, the interaction between the external magnetic field and the generated magnetic field is such that the spacecraft uses the torque rods to generate a dipole that aligns itself with the external magnetic field. However, it will do so slowly and in a damped pendulum motion.

Additionally, the solenoid dipole moments and calculated torque should exhibit a damping behavior as those shown in Figures 5.3 and 5.4. The torque produced was limited to ± 33 mA; therefore, saturation occured at the beginning of the cycle. The torque decayed as the air–bearing corrects its attitude during the dipole alignment process. This behavior is similar to the oscillating decay of the Euler angles measurements and requested dipole moments as shown in Figures 5.1 and 5.3

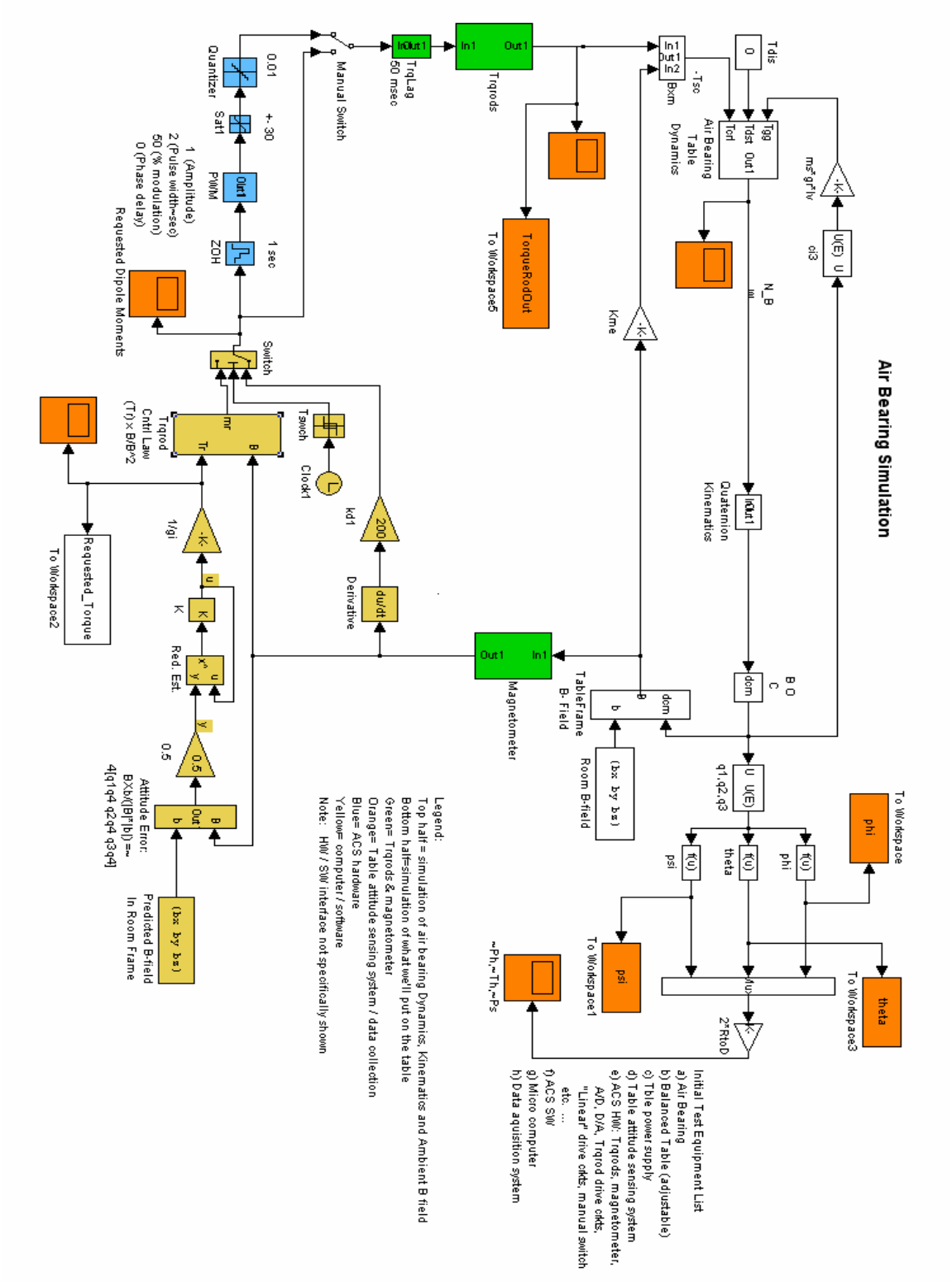


Figure 5.2 NPSAT1 Air–Bearing Simulation Model [Provided by Prof. Barry S. Leonard, NPS SSAG]

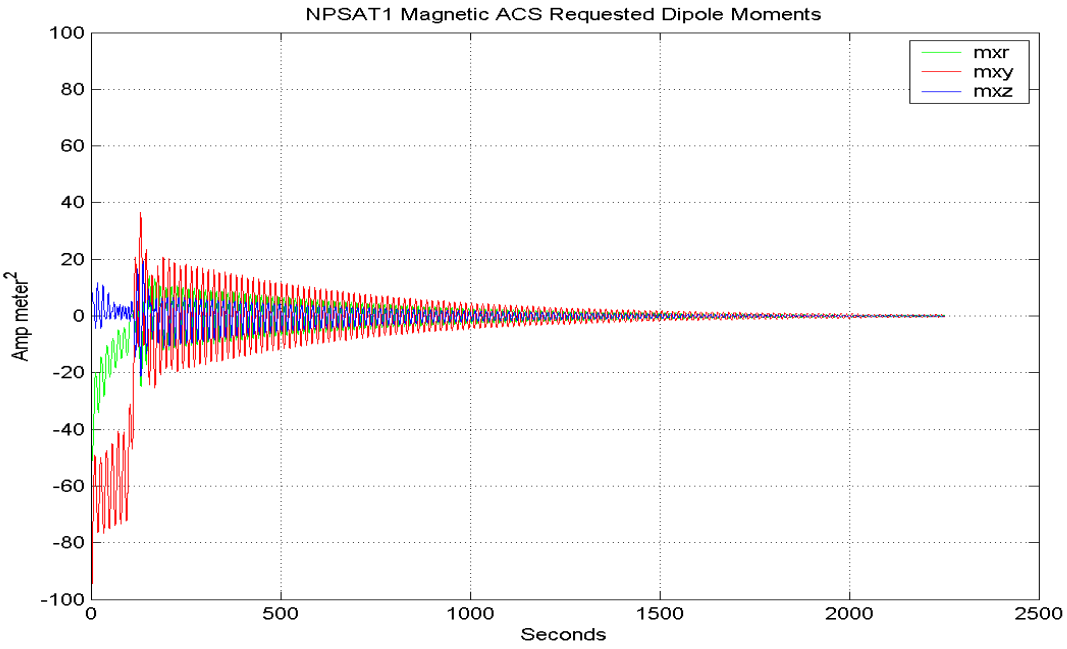


Figure 5.3 NPSAT1 ACS Air–Bearing Requested Dipole Moment Simulation

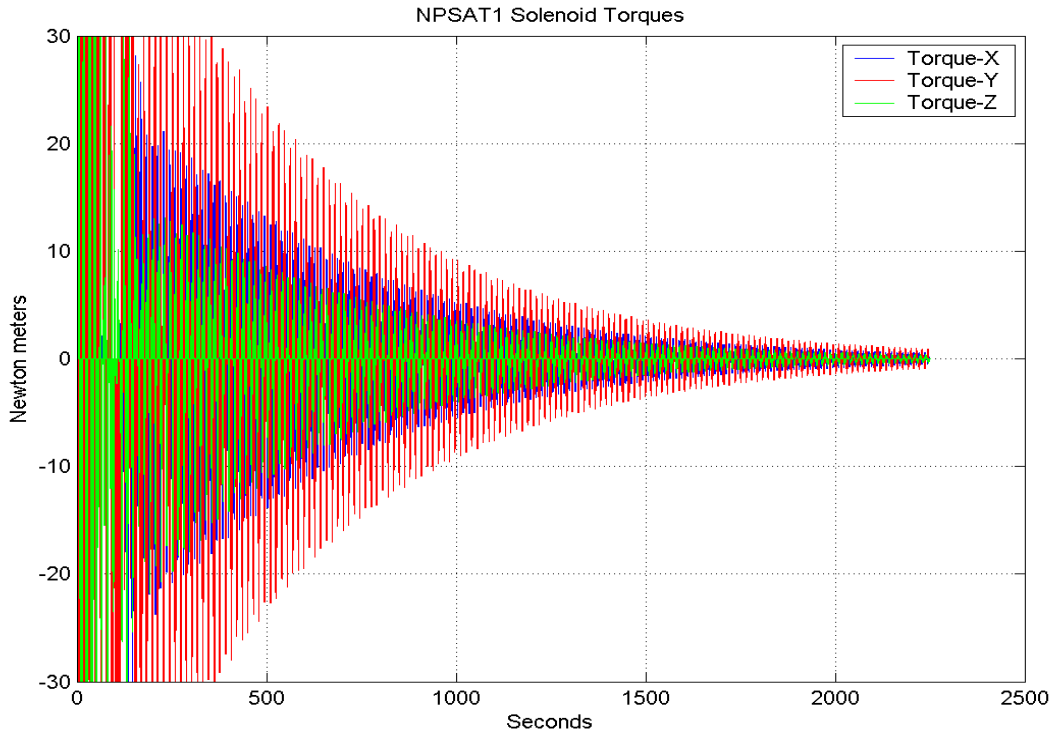


Figure 5.4 NPSAT1 ACS Air–Bearing Torque Rod Ouput Decay Simulation

The abridged static version of the air–bearing ACS model is shown in Figure 5.5. Its purpose was to establish that single magnetic field values could be entered into the model to produce a torque based upon the collected laboratory ephemeris data.

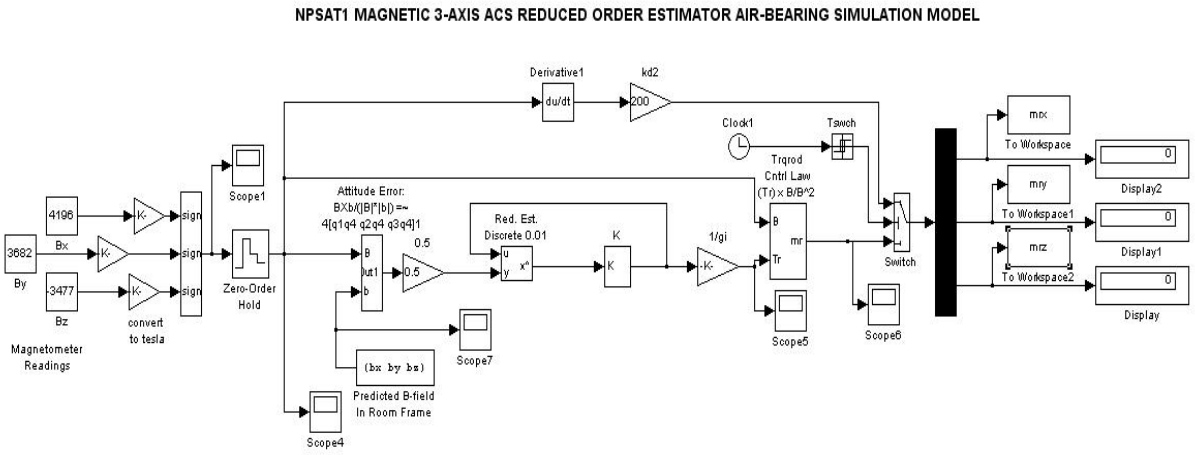


Figure 5.5 NPSAT1 Magnetic ACS Air–bearing Static Plant Model

Static magnetometer and external magnetic values were entered into the simulation model. The resultant dipole moment was on the order of 10^6 . This was an expected condition since the values entered were constant, thus the plant attempted to apply a large dipole moment to align the air–bearing model to the desired frame.

D. NPSAT1 ATTITUDE CONTROL SYSTEM MODEL CONVERSION TO ANSI 'C' CODE USING MATLAB REAL TIME WORKSHOP

The advantage of modeling the ACS using MATLAB and SIMULINK software was that it can also be used to convert the model into ANSI 'C' executable computer code. SIMULINK possesses a feature called Real Time Workshop that converts, compiles, and executes the design model. Once the model was converted, the generated code was used for *hardware-in-the-loop* experiments. Moreover, before, during, and after execution, signals can be modified (*tuned),* traced, and stored for follow on examination. "Generated code can run on PC hardware, DSP's, microcontrollers on bare-board environments, and with commercial or proprietary real time operating systems." [13, p. 1–2]

The ANSI ‘C’ code conversion is completed in four steps. These are

- analysis of the SIMULINK model to ensure there are no unsupported SIMULINK block library functions,
- reading of the SIMULINK model into a “intermediate hierarchical representation called *model.rtw*” [Ref. 14, p. 2–4],
- translation of the *model.rtw* file into ANSI ‘C’ code using a software’s “Target Language Compiler” [Ref. 14, p. 2–4], and
- construction of a *makefile* and compilation into an executable file, *model.exe*.

The Real Time Workshop environment allows signals to be *tuned*. This can occur prior to code generation or during code execution. However, the most valuable feature that this function supplied was the ability to trace a real–time signal throughout the model. This feature came into critical use while debugging the ACS single board computer software and in the analysis of torque rod commands. Further discussion regarding this issue is reserved for the following sections.

The model used in the conversion process is shown in Figure 5.6. As displayed, the system as described in Chapter V, Section B was used with the addition of an RS-232 interface connection. The interface connection functions as the relay device between the air–bearing’s single-board computer, magnetometer, and torque rods. The RS-232 was the input/output conduit that received magnetometer readings and sent torque commands to the torque driver board. The RS-232 interface to the system is shown in Figure 5.6. The RS-232 settings were established as listed in Table 5.1.

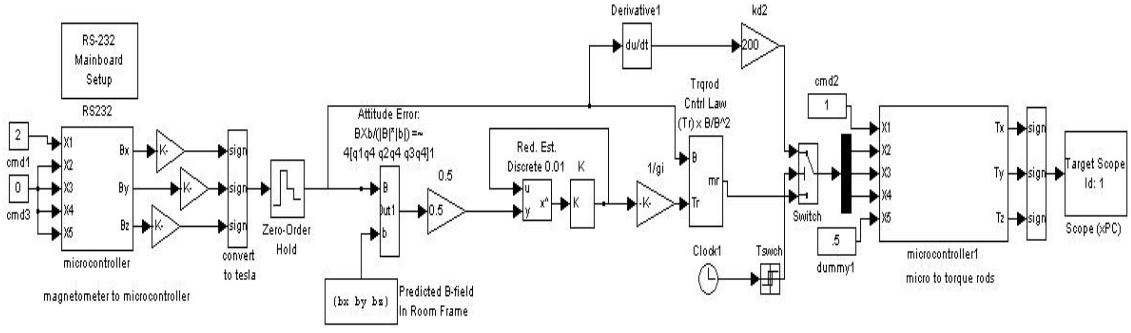


Figure 5.6 NPSAT1 ACS Air–bearing Dynamic Plant

Port	Baud Rate	# Bits	# Stop Bits	Parity	Protocol	Buffer (Send)	Buffer (Rcv)
COM1	9600	8	1	None	None	1024	1024

Table 5.1 RS–232 Parameters

Communication was established between the micro–controller computer and the ACS model using a MATLAB .m file. A copy of the program is located in Appendix K. The MATLAB code used to initialize the channel is given by

```
%Simulink RS232 set up parameters
RS232.SendData = '%d,%d,%d,%d,%d\n';   %d#1=command ; d#2-#4=tx,ty,tz
                                          d#5=time duration
RS232.RecData = '%d,%d,%d\r';          % d#1-3=magnetometer readings,
                                          bx,by,bz
RS232.InputPorts = [1 2 3 4 5];         % identification of serial port block
RS232.OutputPorts = [1 2 3];            % identification of serial port block
RS232.Timeout = 0.005;                  % time out value less than 50 mSec
RS232.EOM = 1;                          % end of message command.
```

The *hardware-in-the-loop* process was established using another SIMULINK feature. Specifically, XPC Target and XPC Host functions were used to simulate the ground station and spacecraft control. After the ACS model was built and compiled using Real Time Workshop, the executable code was loaded to the air–bearing's micro–controller computer. The air–bearing computer was identified as the Target while a desktop personal computer served as the Host or ground station. The two computers communicated via serial lines; however, upon completion of preliminary tests that proved all hardware and software components functioned properly, a remote system will be incorporated. The manner in which the magnetometer, power supplies, torque driver board, and Host and Target computers were connected is shown in Figure 5.7. Additionally, during remote operations, the air–bearing uses batteries to supply electricity vice using a DC power supply. The power supply provided 30 V DC to power the magnetometer, torque driver board, and micro–controller computer.

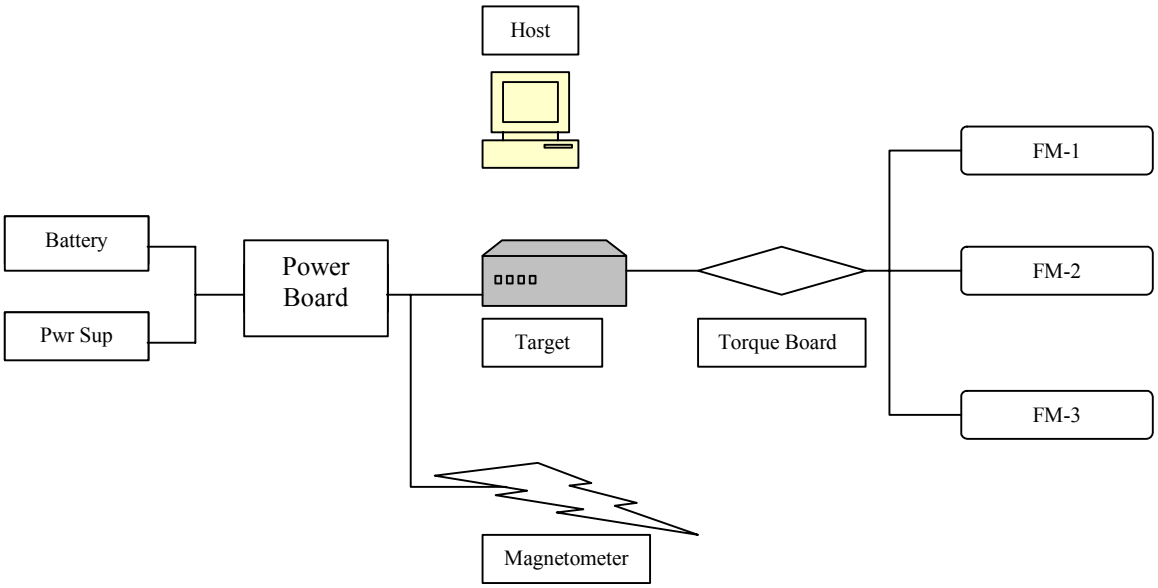


Figure 5.7 NPSAT1 ACS Air–Bearing Control Configuration

External magnetic field ephemeris data that was stored by the micro–controller computer is listed in Table 5.2. This data was collected during the third laboratory magnetic field surveys as discussed in Chapter IV. All values are in Tesla.

Following the successful conversion from a model (.mdl) file to an executable (.exe) file, *hardware-in-the-loop* tests commenced. Initial trials were encouraging; however, a problem with the magnetometer signal routing was discovered. Initially, magnetometer readings were received through the RS-232 and successfully passed to the ACS for calculation. However, after repeated experiments designed to demonstrate stability, it was discovered that the output data was not reliable. Therefore, Real Time Workshop target scopes were used to analyze the different signals present throughout the air–bearing plant. The expected result was a dipole moment command sent through the RS-232. However, after repeated experiments, the output expected from the ACS did not agree with the theoretical function of the ACS. At first, it was believed there was a sequencing problem within the SIMULINK model. Simulation times, signal parameters, and data collection properties were adjusted to determine the source of the problem. Two sets of tests successfully built the model into ANSI 'C' code, accepted magnetometer readings, and calculated dipole moment values. However, this was the exception

vice the norm during the first stages of testing. Thereafter, each time the model was rebuilt, the signals exhibited different and uncorrelated behavior. Each section of the ACS executable model, RS-232 parameters, and serial communication between the Target and Host computers were reexamined. No problems were discovered. Therefore, attention was turned to the torque driver circuit board and micro–controller software. The next section discusses the faults discovered with the torque driver circuit board and micro-controller.

90º Inclination B Field (Tesla)			75º Inclination B Field (Tesla)			105º Inclination B Field (Tesla)		
B_x	B_y	B_z	B_x	B_y	B_z	B_x	B_y	B_z
8.72E-05	-4.42E-05	-2.73E-05	8.72E-05	-4.42E-05	-2.73E-05	9.60E-05	-3.42E-05	-4.15E-08
8.59E-05	-1.09E-05	-5.20E-05	8.59E-05	-1.09E-05	-5.20E-05	0.0001	-1.21E-05	-9.65E-06
8.44E-05	1.66E-05	-5.17E-05	8.44E-05	1.66E-05	-5.17E-05	9.91E-05	1.71E-05	-9.23E-06
8.27E-05	3.93E-05	-3.90E-05	8.27E-05	3.93E-05	-3.90E-05	9.19E-05	4.02E-05	3.50E-06
8.22E-05	5.42E-05	-9.38E-06	8.22E-05	5.42E-05	-9.38E-06	8.10E-05	5.38E-05	2.27E-05
8.17E-05	5.28E-05	1.93E-05	8.17E-05	5.28E-05	1.93E-05	6.75E-05	4.96E-05	5.50E-05
8.24E-05	3.78E-05	4.34E-05	8.24E-05	3.78E-05	4.34E-05	5.77E-05	3.26E-05	7.67E-05
8.36E-05	1.38E-05	5.65E-05	8.36E-05	1.38E-05	5.65E-05	5.30E-05	7.74E-06	8.76E-05
8.53E-05	-1.31E-05	5.56E-05	8.53E-05	-1.31E-05	5.56E-05	5.43E-05	-1.59E-05	8.66E-05
8.68E-05	-3.74E-05	4.04E-05	8.68E-05	-3.74E-05	4.04E-05	6.27E-05	-4.22E-05	7.08E-05
8.79E-05	-4.99E-05	1.58E-05	8.79E-05	-4.99E-05	1.58E-05	7.41E-05	-5.33E-05	4.80E-05
8.75E-05	-4.99E-05	-1.43E-05	8.75E-05	-4.99E-05	-1.43E-05	8.58E-05	-5.03E-05	2.49E-05

Table 5.2 NPSAT1 ACS Air–Bearing External Magnetic Field Vector Component Ephemeris Values (Tesla)

E. NPSAT1 AIR-BEARING SOLENOID AND TORQUE DRIVER CIRCUIT BOARD TESTS

Upon completion of the initial battery of tests, attention was focused upon the air–bearing's hardware. The first point of examination was the solenoids and torque driver board.

A test plan was developed to isolate problems. The plan included the performance of current measurements to each solenoid. These measurements would verify that each solenoid coil was receiving the proper amount of current.

The second test examined whether the torque driver circuit board was working as designed. Specifically, the torque driver should receive a digital signal from the micro–controller and convert it into an analog signal. The analog signal represents the amount of current that is applied to the solenoid. This current represents the amount of dipole moment calculated by the ACS.

However, before the test results and the solutions proposed are discussed, it is necessary to discuss in more detail the solenoid's and torque driver circuit board's characteristics.

As discussed in Chapter IV, NPSAT1 will use three solenoids (torque rods) to produce a dipole moment. Each coil will produce approximately a dipole moment, m, of $\pm 33\ \mathrm{A}\cdot\mathrm{m}^2$ from an maximum input of 145 mA of current. The manufacturer established the dipole moment values for each coil through tests results. [8] However, it was desirable to know the magnitude of the coil's generated magnetic field at discrete points away from the coil, the number of coils used, the flux generated, and the resultant torque expected from each coil. Therefore, based upon the dipole moment and maximum current, the coils magnetic field were calculated using [14]

$$B(Z) = \frac{\mu_0}{2\pi}\frac{NiA}{Z^3} = \frac{\mu_0}{2\pi}\frac{m}{Z^3} \tag{5.1}$$

where

Z = distance normal to coil,

$\mu_0 = 4\pi\times 10^{-7}\ \ \mathrm{T}\cdot\mathrm{m/A}$ = permeability constant,

N = number of coil wire turns,

i = current $(\mathrm{Amperes})$, and

$m = NiA$.

To use Equation 5.1, the number of coils and area enclosed by each coil first needed to be determined. The number of coils was found using $N = m/iA$. [14] The number of wire wraps per coil, N, was determined to be 3.9749×10^5 wraps. This value is an approximation since the wrapping method, wire gage, and core material used during construction are proprietary infor-

mation and, therefore, unknown. However, the calculations were believed to be a satisfactory approximation.

Each solenoid's flux $\left(\text{Tesla} \cdot \text{m}^2 = \text{Weber}\right)$ was approximated by $\Phi = BA = \mu_0 inA.$.

[14] The coil inductance $\left(\text{Tesla} \cdot \text{m}^2 / \text{Amp} = \text{Henry}\right)$ was found by $L = N\Phi / i$. [14] The computed magnetic field, flux, inductance, and torque approximations generated at an orthogonal distances of 0 to 1 meter from the coil are listed in Table 5.3. As expected, the plot of flux vs. distance, Figure 5.8, displays an exponential decay of the magnetic field as it traveled further away from the source.

Distance (Meters)	Magnetic Field, B (Tesla)	Flux, Φ (Weber)	Inductance, L (Henry)	Torque, T (Newton Meter)
0	3.29E-05	1.88E-08	0.051655	0.001086
0.05	1.17E-06	6.67E-10	0.00183	3.847e-005
0.10	2.36E-07	1.35E-10	0.000371	7.8019e-006
0.15	9.48E-08	5.43E-11	0.000149	3.1288e-006
0.20	5.03E-08	2.88E-11	7.89E-05	1.6587e-006
0.25	3.10E-08	1.77E-11	4.86E-05	1.0216e-006
0.30	2.09E-08	1.20E-11	3.28E-05	6.9066e-007
0.35	1.51E-08	8.63E-12	2.37E-05	4.9747e-007
0.40	1.14E-08	6.51E-12	1.78E-05	3.7512e-007
0.45	8.87E-09	5.08E-12	1.39E-05	2.9283e-007
0.50	7.12E-09	4.08E-12	1.12E-05	2.3488e-007
1.0	9.71E-10	5.56E-13	1.52E-06	3.2056e-008

Table 5.3 NPSAT1 Air–Bearing Solenoid Electrical Properties

The torque driver circuit board was designed to receive a digital value from the micro–controller computer. The digital value represents the dipole moment calculated by the ACS plant. The digital input was converted to an analog value. This analog signal represents the amount of current sent through a coil and is the output of the torque driver circuit board.

Regulating the coil current regulates the amount of torque produced by the torque rod. The current ranges from 0 to 145 mA. An AD7224 8-bit digital-to-analog (DAC) converter was used in the circuit board. The reference voltage used for the DAC was 5 V DC. There-

fore, the incoming digital signal was quantized to 256 analog levels. The analog signal was

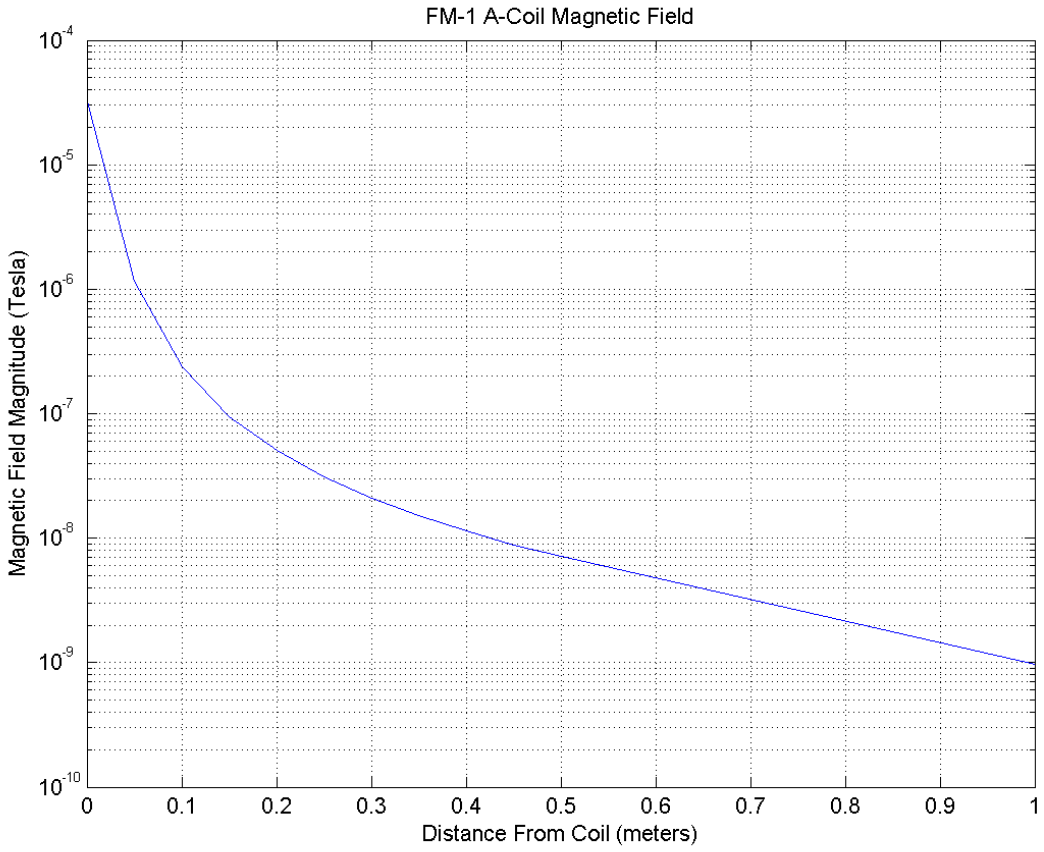


Figure 5.8 NPSAT1 Torque Rods Calculated Induced Magnetic Field

multiplied by 25 V DC which produces the output. The 25 V DC scaler was set by two operational amplifiers. The first was used to boost the signal, while the second was used to buffer the output. The output voltage was then converted into current using Ohms Law, $i = V/R$. The coil resistance was denoted by R and was equal to 172 ohms. [8]

Quantization values of the digital-to-analog current values for the AD7224 and an AD565A 12-bit DAC are presented in Table 5.4. The values represent binary levels, 2^n, where n = 0,1,2,3,4,5,6,7. The torque driver board schematic is displayed in Figure 5.9.

Torque driver circuit board tests were accomplished by inducing and measuring current sent to the solenoids. Specifically, a Tektronix TDS 2024 Digital Oscilloscope, a Tektronix

8-Bit DAC	Current (AMPS)	12-Bit DAC	Current (AMPS)
00000000	0	000000000000	0
00000001	0.00056777	000000000001	0.000035486
00000010	0.0011355	000000000010	0.000070971
00000100	0.0022711	000000000100	0.00014194
00001000	0.0045422	000000001000	0.00028388
00010000	0.0090843	000000010000	0.00056777
00100000	0.018169	000000100000	0.0011355
01000000	0.036337	000001000000	0.0022711
10000000	0.072674	000010000000	0.0045422
11111111	0.14478	000100000000	0.0090843
		001000000000	0.018169
		010000000000	0.036337
		100000000000	0.072674
		111111111111	0.14531

Table 5.4 NPSAT1 Torque Driver Digital-to-Analog Conversion

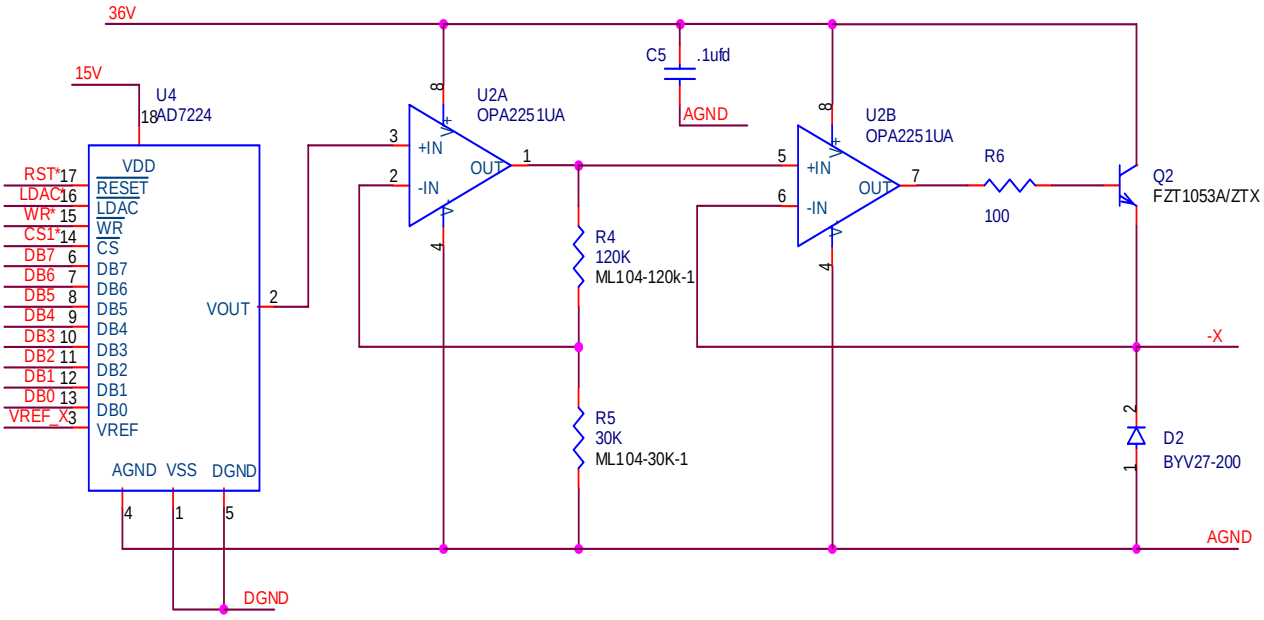


Figure 5.9 NPSAT1 Torque Driver Circuit Board DAC [Provided by Mr. Ron Phelps, NPS SSAG]

AM503B AC/DC Current Probe Amplifier, and a Tektronix A6302 20 Ampere AC/DC Current Probe were used as test equipment. The current probe was degaussed before each measurement and set in-line with the current flow. Torque request values of 255, 128, 64, 32, 16, 8, 4, 2, 1, 0 were entered via a MATLAB program. The expected result was for the microcontroller computer to relay the digital request to the torque driver board, convert it to an analog format, and apply the appropriate current.

Figure 5.10 is a sample of the results. Use of the current probe amplifier required the oscilloscope to always be set to 10 mV per division. Depending on the current size, the current probe amplifier voltage scale was set to 20, 10, 5, 2, or 1 mV per division. For example, Figure 5.10 displays a pulse with an amplitude of 36.4 mV. The current probe amplifier range was set to 20 mV per division. Thus, the pulse amplitude was twice that represented on the oscilloscope display and the actual measurement equaled 72.8 mV. The voltage measured had a one-to-one conversion from voltage to current, 1 mV = 1 mA. Furthermore, all pulse widths were 2.5 seconds. Prior to each measurement, the system noise was measured and treated as a bias. The bias was subtracted from the pulse amplitude.

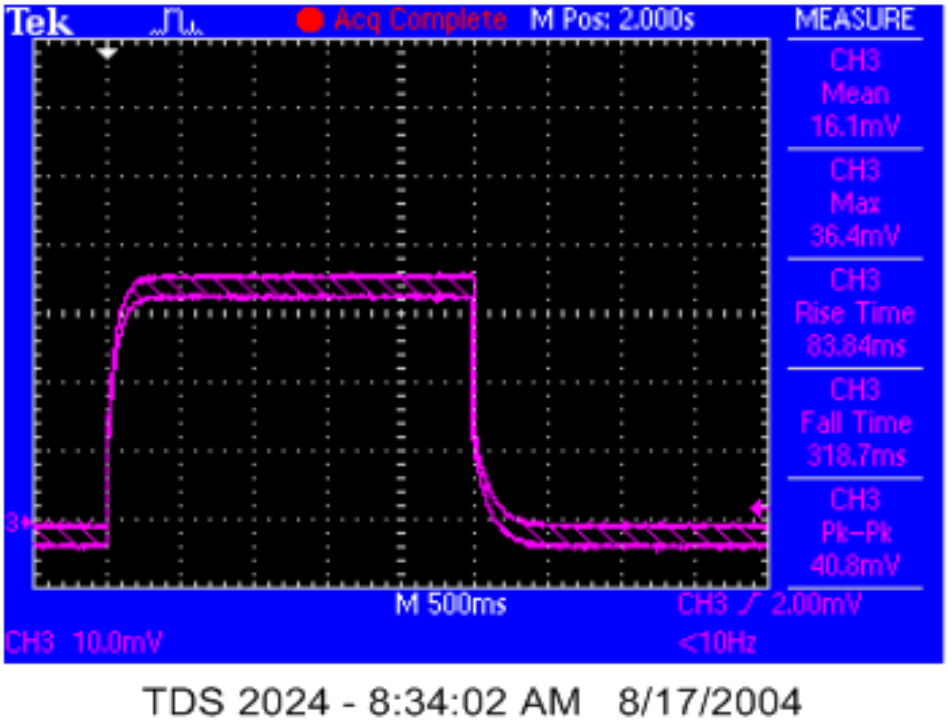


Figure 5.10 NPSAT1 Torque Driver Circuit Board Current Test (Bit Level 128)

The first test using a value of 255 was successful. However, as the bit values were sequentially stepped down, the current levels did not respond. These tests were completed for each solenoid. All coils responded in the same fashion with the exception of the negative *Z* coil. The *Z* coil did not respond whatsoever. Upon investigation, it was discovered that two port pins to the torque driver circuit board and a wire to the negative *Z* coil were broken. The pin faults prevented any quantization below the bit value 255. The broken wire prevented any current to be applied to the negative *Z* coil. The circuit board pins and wire connection were repaired. The D/A conversion tests were accomplished upon completing the repairs.

The D/A conversion process did not follow a linear path as anticipated. The step size for the 8-bit DAC is approximately 0.57 mA. A review of Tables 5.5 to 5.7 initially proved that the digital-to-analog conversion progress did not function well. Errors greater than 0.7% were deemed unacceptable based upon the DAC step size.

Errors were determined by

$$\text{Error} = \left| \frac{\text{Expected Value} - \text{Measured Value}}{\text{Expected Value}} \right| \times 100. \tag{5.2}$$

Bit Level	Expected Value (mA)	+X (mA)	% Error	−X (mA)	% Error
255	140	141.6	1.1%	141.8	1.3%
128	70	70.4	0.6%	68.5	2.1%
64	35	34.6	1.1%	34.6	1.1%
32	17.5	16.8	4.0%	17.1	2.3%
16	8.75	7.2	17.7%	8.7	0.6%
8	4.375	4.4	0.6%	4.78	9.3%
4	2.1875	2.04	6.7%	2.45	12.0%
2	1.09375	1.12	2.4%	1.05	4.0%
1	0.546875	0.4	26.9%	0.475	13.1%

Table 5.5 NPSAT1 X–Torque Rod Command Digital-to-Analog Conversion Test 1

Bit Level	Expected Value (mA)	+Y (mA)	% Error	−Y (mA)	% Error
255	140	140.00	0.0%	144	2.9%
128	70	69.00	1.4%	72.14	3.1%
64	35	33.00	5.7%	35.5	1.4%
32	17.5	15.82	9.6%	16.57	5.3%
16	8.75	7.12	18.6%	7.61	13.0%
8	4.375	4.15	5.1%	3.53	19.3%
4	2.1875	1.65	24.6%	2.138	2.3%
2	1.09375	1.06	3.1%	1.16	6.1%
1	0.546875	0.40	26.9%	0.588	7.5%

Table 5.6 NPSAT1 Y–Torque Rod Command Digital-to-Analog Conversion Test 1

Bit Level	Expected Value (mA)	+Z (mA)	% Error	−Z (mA)	% Error
255	140	142.8	2.0%	143.5	2.5%
128	70	72.2	3.1%	71.3	1.9%
64	35	35	0.0%	35.7	2.0%
32	17.5	17.23	1.5%	16.6	5.1%
16	8.75	8.6	1.7%	8.92	1.9%
8	4.375	4.1	6.3%	4.46	1.9%
4	2.1875	1.65	24.6%	1.885	13.8%
2	1.09375	0.684	37.5%	1.01	7.7%
1	0.546875	0.4	26.9%	0.6	9.7%

Table 5.7 NPSAT1 Z–Torque Rod Command Digital-to-Analog Conversion Test 1

Since the first set of current values was different from the expected results, a second set of data points was recorded. The results are displayed in Tables 5.8–5.10. Just as seen in the DAC Test 1 results, Test 2 results show unacceptable quantization error.

Ideally, the DAC process should follow a linear step representation. It was quite clear from the randomness of the error seen in Tables 5.5–5.10 that noise was introduced that grossly distorted the analog signal. This noise was of a major concern because it had a disruptive effect upon the system that needed to respond precisely to fine torque commands. The distortion was most prevalent in the lower bit values as witnessed by the high error percentages.

Bit Level	Expected Value (mA)	+X (mA)	% Error	−X (mA)	% Error
255	140	144.4	3.1%	147.4	5.3%
128	70	70.48	0.7%	74.6	6.6%
64	35	36.73	4.9%	34.8	0.6%
32	17.5	18.44	5.4%	16.4	6.3%
16	8.75	9.2	5.1%	8.12	7.2%
8	4.375	3.64	16.8%	4.06	7.2%
4	2.1875	2.16	1.3%	2.051	6.2%
2	1.09375	0.94	14.1%	0.87	20.5%
1	0.546875	0.56	2.4%	0.68	24.3%

Table 5.8 NPSAT1 X–Torque Rod Command Digital-to-Analog Conversion Test 2

Bit Level	Expected Value (mA)	+Y (mA)	% Error	−Y (mA)	% Error
255	140	141.7	1.2%	142.1	1.5%
128	70	76.8	9.7%	71.7	2.4%
64	35	39.8	13.7%	35.9	2.6%
32	17.5	16.6	5.1%	16.9	3.4%
16	8.75	7.35	16.0%	8.22	6.1%
8	4.375	4.08	6.7%	3.82	12.7%
4	2.1875	1.68	23.2%	2.47	12.9%
2	1.09375	1.64	49.9%	1.42	29.8%
1	0.546875	no read-ing	no value	0.72	31.7%

Table 5.9 NPSAT1 Y–Torque Rod Command Digital-to-Analog Conversion Test 2

Bit Level	Expected Value (mA)	+Z (mA)	% Error	−Z (mA)	% Error
255	140	145.9	4.2%	146.24	4.5%
128	70	75.8	8.3%	75.04	7.2%
64	35	36.52	4.3%	37.44	7.0%
32	17.5	18.5	5.7%	19.6	12.0%
16	8.75	8.73	0.2%	9.4	7.4%
8	4.375	3.99	8.8%	4.77	9.0%
4	2.1875	1.86	15.0%	2.47	12.9%
2	1.09375	1.08	1.3%	1.47	34.4%
1	0.546875	0.89	62.7%	0.925	69.1%

Table 5.10 NPSAT1 Z–Torque Rod Command Digital-to-Analog Conversion Test 2

This noise phenomenon is exhibited in Figure 5.11. Some noise distortion was evident since the pulse's amplitude does not remain constant.

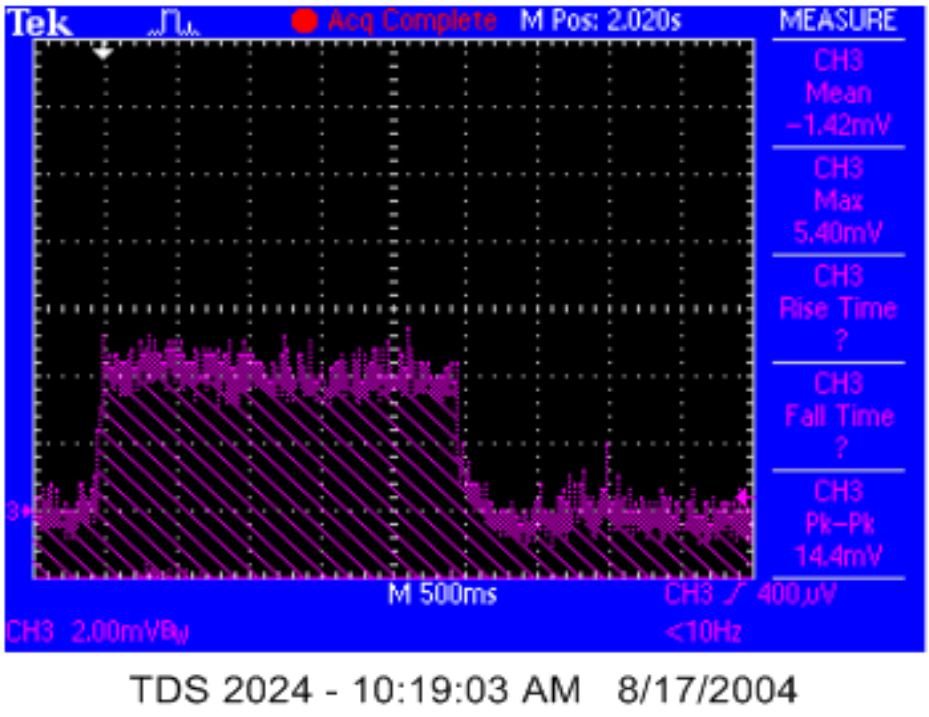


Figure 5.11 NPSAT1 Torque Driver Circuit Board Current Test (Bit level 1)

The torque driver circuit board current distortions could be caused by three sources of noise.

The first hypothesis was that the instrumentation was introducing noise. This statement was correct. The instrumentation created noise; however, it was accounted for in the measuring process. As discussed previously, the mean instrumentation noise was measured prior to each recorded data point. The mean noise value was subtracted from the pulse amplitude which produces the resultant signal.

The second theory focuses upon the torque rod driver circuit board. Noise measurements of the circuit board's components yielded resident noise levels as high as 40 mV.

The third theory was that the current values used to induce the coils were too low and would mix with the systems resident thermal noise, thus, becoming distorted and unreliable.

Ideally, at the lower bounds of the torque and current levels, the drive circuit should follow the process linearly. However, a review of the manufacturers torque rod test sheets revealed a non-linearity of the current–to–torque output begins to falter as it approaches the origin. This phenomenon is exhibited in Figure 5.11. The distortion occurs at approximately ±50 mA. This point equals to approximately a bit value of 91. Therefore, based upon the manufacturer's data shown in Figure 5.12, two-thirds of the current level demands will go unaltered while the other one-third was susceptible to noise distortion. This phenomenon is exhibited in the DAC current tables shown above. Bit values greater than 64 show a tendency to possess only a small amount of error. However, once past the 64-bit value, error percentages rise quickly. Therefore, alternatives to the current hardware process must be addressed.

Three possible methods are proposed to alleviate this issue. The first is to use a DAC that quantizes the current levels to a finer resolution. Thus, the reason for presenting the 12-bit DAC values in Table 5.4. A quantization step size of 35 μA will provide extreme resolution, vice using a current step size of 0.5 mA. Furthermore, instead of having only 256 current divisions, the 12-bit DAC provides 4096 current divisions. However, it is theorized that the solenoid's core material will still contain residual magnetism. Moreover, since all electronic components do not behave linearly as they approach the finest resolution, it was postulated that the 12-bit DAC may not provide the necessary resolution. Therefore, since the DAC conversion process is non–linear in the lower bounds, using a non-linear quantization scheme or a

companding amplifier may provide resolution. Companding is mostly used in the telecommunications industry to boost a faint analog signal. Specifically, a compression

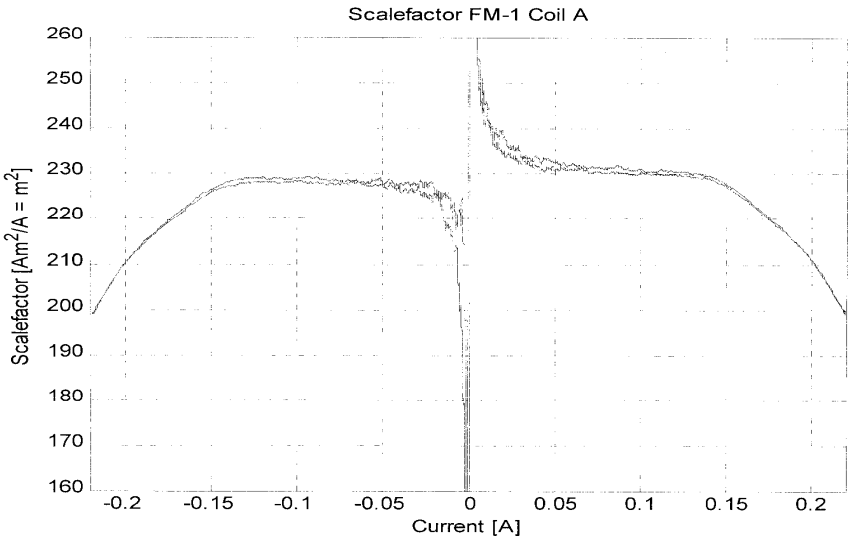


Figure 5.12 NPSAT1 Solenoid Current Plot [From Ref. 8.]

amplifier compresses an analog signal. This compressed signal is uniformly quantized using a companding law such as that given by [15]

$$F(s) = \operatorname{sgn}(s)\frac{\ln(1+\mu|s|)}{\ln(1+\mu)} \tag{5.2}$$

where

μ = bit value, and

s = uncompressed analog signal magnitude.

The resultant quantization scheme takes on a logrithmic nature where lower values are quantized to a finer resolution while higher values can be quantized using less steps, thus providing fidelity while reducing quantization noise or quantization error. Development and implementation of a companding scheme was deemed as a topic for further research.

The second option is to add or to replace the solenoids with air coils. Air coils will generate the same amount of dipole moment but will not exhibit the properties that the aluminum alloy core solenoids exhibit at lower current levels. The MICROCOSM core permeability

value was unavailable due to proprietary reasons. However, using an air core may solve the non-linearity effect since the material permeability property of air is equal to one. Thus, the solenoid core will not retain a magnetic signature and cause disturbances at the lower bounds. Investigation in this matter is deemed most important and has been listed as an issue for further research.

Preliminary investigations revealed that an air coil that would meet the project's goals could be constructed. Initial coil construction could be based upon Table 5.11. Specifically, the table identifies 16 different wire gage sizes, areas, required turns, resistances, current, dipole moments, magnetic fields, flux, inductances, and size parameters. This table was created in cooperation with NPS Professor Barry S. Leonard. The EXCEL spreadsheet can be adjusted to accommodate different coil core lengths, diameters, voltages, currents, or any of the other specifications listed.

The third option is to reengineer the torque driver circuit board. This process involves research to determine the best solid-state components that reduce noise while simultaneously offering fine current resolution. Currently, no matter how fine the quantization scheme is, the 40 mV of noise present within the circuit board will still cause masking. An additional possibility is to add a filter or filters to the current driver board that would isolate the noise from the signal desired. The identification of the filter type and possibility of reengineering the circuit board is considered an additional research topic.

F. NPSAT1 TORQUE ROD MAGNETIC FIELD MEASUREMENTS

The NPSAT1 air–bearing tests began with an assumption that all hardware and software components were working properly and as designed. However, following the initial *hardware-in-the-loop* testing, it was deemed that further testing to the air–bearing components was required prior to reinitializing the *hardware-in-the-loop* experiments. Therefore, this section will discuss the results found from measuring the induced magnetic field in the three torque rods.

The tests conducted were performed on the positive axis coils only. The process was accomplished using a MATLAB program designed to induce a triggered current based upon the integer bit values of 2^n, where n = 1,2,3,4,5,6,7. Thus, the bit values 255, 128, 64, 32, 16,

Rod Length		Lr~in	10		Lr~cm	25.4			PermConstant(μ)	1.26E-06	T*m/amps
Rod Diameter		Dr~in	1		Dr~cm	2.54	D~m	0.0254	r = Rod Radius(m)	0.0127	
k=Total D /Wire D			1.2						A= Rod Area (m^2)	5.07E-04	
Voltage~v			28								
Resistivity of Copper		rho~ohm.cm	1.72E-06								
B&S Gauge	tc~cm	ttotal~cm	N	Lw~cm	R~ohms	i~amps	M~amp.m^2	B~Tesla	z=dist from coil center (m)	0.5	
		k*tc	Lr/ttotal	N*pi*Dr	Lw*rho/Aw	V/R	i*A*N	μ*N*I*A/2*pi*z^3	B in coil~Tesla	BA	F=flux~Tm^2	L=ind~Tm^2/amps
10	0.2588	0.31	32.20	256.94	0.0084	3325.0725	54.2516	5.37E-05	2.20E-03	1.12E-06	1.08E-08	N*F/I
12	0.2053	0.25	40.59	323.90	0.0169	1669.8741	34.1399	3.38E-05	1.75E-03	8.85E-07	2.16E-08	
14	0.1628	0.20	51.19	408.46	0.0338	827.6972	21.4681	2.12E-05	1.38E-03	7.02E-07	4.34E-08	
16	0.1291	0.15	64.55	515.08	0.0678	412.7500	13.5001	1.34E-05	1.10E-03	5.56E-07	8.70E-08	
18	0.1024	0.12	81.38	649.39	0.1359	205.9720	8.4935	8.41E-06	8.71E-04	4.41E-07	1.74E-07	
20	0.08118	0.10	102.65	819.13	0.2728	102.6255	5.3390	5.28E-06	6.90E-04	3.50E-07	3.50E-07	
22	0.06438	0.08	129.44	1032.88	0.5470	51.1872	3.3673	3.32E-06	5.48E-04	2.77E-07	7.02E-07	
24	0.05106	0.06	163.21	1302.33	1.0965	25.5359	2.1118	2.09E-06	4.34E-04	2.20E-07	1.41E-06	
26	0.04049	0.05	205.81	1642.31	2.1989	12.7336	1.3279	1.31E-06	3.44E-04	1.74E-07	2.82E-06	
27	0.03606	0.04	231.10	1844.07	3.1130	8.9947	1.0533	1.04E-06	3.07E-04	1.55E-07	3.99E-06	
28	0.03211	0.04	259.52	2070.91	4.4089	6.3508	0.8362	8.26E-07	2.73E-04	1.38E-07	5.65E-06	
30	0.02546	0.03	327.31	2611.82	8.8445	3.1658	0.5251	5.20E-07	2.17E-04	1.10E-07	1.13E-05	
32	0.02019	0.02	412.75	3293.56	17.7354	1.5788	0.3302	3.27E-07	1.72E-04	8.70E-08	2.27E-05	
34	0.01601	0.02	520.51	4153.47	35.5693	0.7872	0.2076	2.06E-07	1.36E-04	6.90E-08	4.56E-05	
36	0.0127	0.02	656.17	5235.99	71.2588	0.3929	0.1306	1.29E-07	1.08E-04	5.47E-08	9.14E-05	
40	0.00799	0.01	1043.36	8325.66	286.4829	0.0977	0.0517	5.11E-08	6.79E-05	3.44E-08	3.67E-04	

Table 5.11 NPSAT1 Sample Air–Coil Parameters

8, 4, 2, and 1 that correspond to their respective current values were used. The program used is presented below. This MATLAB program was drafted by Mr. James Horning, an NPSAT1 staff engineer. The program input argument, *t*, was changed for each new measurement. The value *t* relates to the bit value desired. Each test was triggered for 2.5 seconds. Selection of which coil to trigger was based upon the second, third, and fourth term of the variable declared as "s = sprintf('1,%d,0,0,%d\n', t, DURATION)." These terms represent the *X*, *Y*, and *Z* coils, respectively. Specifically, the %d term represents the excited coil. The number one was a command signal. A zero commands a coil not to excite. The fifth value represented the time duration in milliseconds that the coil was triggered.

```
 % NPSAT1 Air-Bearing Torque Rod Driver Algorithm
% drafted by Mr. James Horning, NPS, Monterey, CA
% 16 August 2004
clear all, clc
s1=serial('COM1', 'BaudRate', 9600, 'Terminator', 13);
fopen(s1);
t = 255;
N = 1;
DURATION = 2500;
for i=1:N,
s = sprintf('1,%d,0,0,%d\n', t, DURATION);
fprintf(s1, s);
pause(DURATION/1000);
s = fgets(s1)
end;
fclose(s1);
delete(s1);
```

The program was run on a PC with the magnetometer connected via a serial port. The magnetometer readings were recorded using a laptop computer. The data recording program used was the same as that used during the laboratory magnetic field surveys discussed in Chapter IV. The magnetometer was centered and set directly against the torque rod. The magnetometer *X*, *Y*, and *Z* axes were aligned to be the same as the torque rod alignment on the air bearing. Even though the magnetometer was set against the torque rod, the actual sensor was ¼ of an inch away from the solenoid. Prior to coil excitation, the magnetic field was measured and recorded for approximately ten seconds. Following this measurement and as the data recording continued, the positive coil was excited. The measurement was continued after coil

excitation to record any residual magnetic affects. Once all measurements were complete for each positive torque rod, a MATLAB program was used to analyze the collected data.The program used to perform this analysis is located in Appendix L. Specifically, the program loads the raw data, extracts the magnetic north pole value, and converts the data from *counts* to Tesla using the relationship exhibited in Equation (4.2). Moreover, the magnetic field vectors were calculated by subtracting the static magnetic field vectors from the apparent induced magnetic field vectors. The resultant vectors represent the induced magnetic field vectors. The magnitude of the induced magnetic field was calculated by taking the square root of the sum of the squares. This was the same relationship as shown in Equation (3.9). Furthermore, once the field magnitude is determined, the program is used to calculate the torque given by $T = \mu B \sin(\theta)$. [14] The variable T is the torque $(\mathrm{N}\cdot\mathrm{m})$, m is the dipole moment $\left(\mathrm{A}\cdot\mathrm{m}^2\right)$, B is the induced magnetic field (Tesla), and θ is the angle between the magnetometer and the solenoid. During these experiments, the angle used was 90°. The dipole moment used was found from the ideal current value that coincides with the respective bit value torque command as noted in the previous section. The dipole moment was found using $m = NiA$**.** [14].

The torque–rod induced magnetic–field measurements results are shown in Figures 5-13 to 5-27. The figures display the generated magnetic field and the associated torque created for its respective bit level. Additionally, Figures 5-16, 5-21, and 5-27 display a chart that plots all of the torque profiles. The purpose of displaying this chart was to show that the torque values decrease as the bit value decreases, thus proving that the torque rods respond to commands. Furthermore, a sample of the coil's residual magnetism was recorded to prove that it exists. This residual effect is plainly seen in Figure 5-28. Specifically, during the last three seconds during data capture, the field was erratic at the lower levels. This occurs because the field had decayed, but the material properties of the coil still maintained a magnetic signature. The residual magnetism effect could cause the spacecraft to incorrectly measure the external magnetic field. This erroneous reading could cause the ACS to respond to an erroneous signal and, thus, an incorrect attitude correction results.

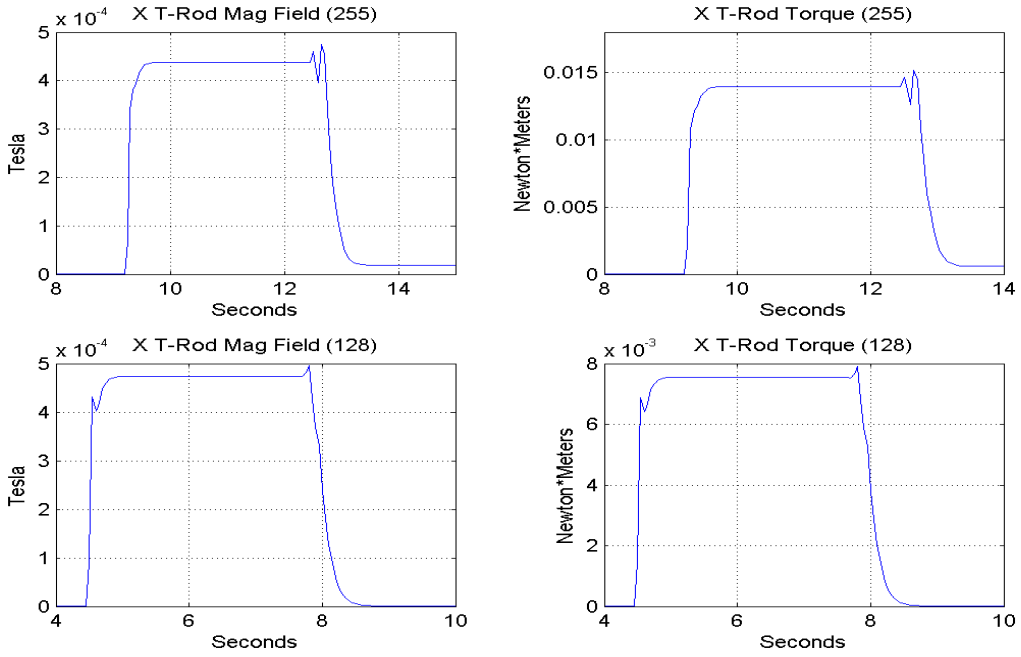


Figure 5.13 NPSAT1 Air–Bearing X Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 255 & 128)

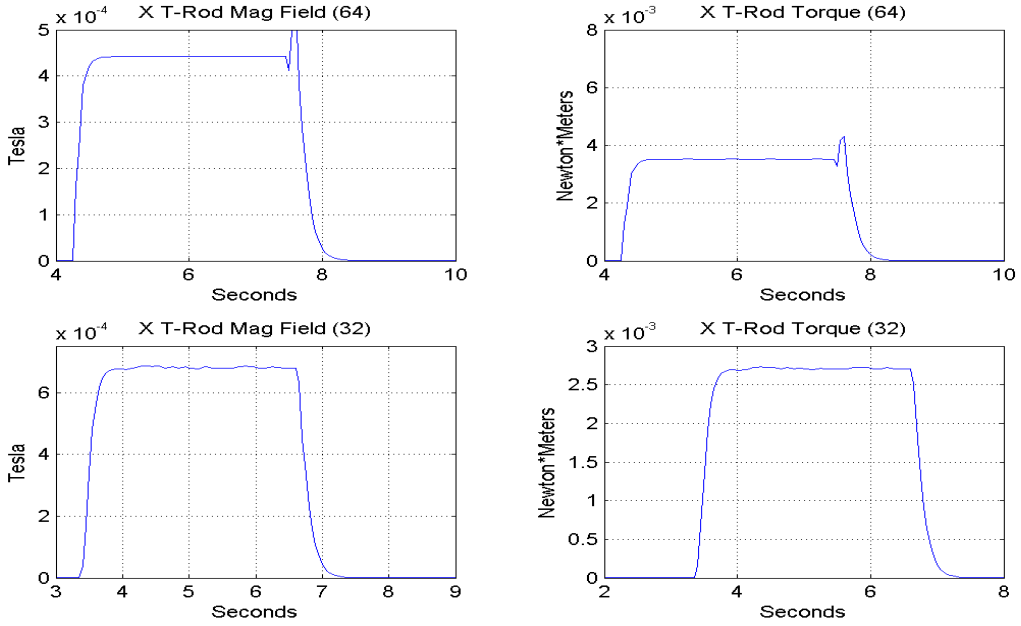


Figure 5.14 NPSAT1 Air–Bearing X Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 64 & 32)

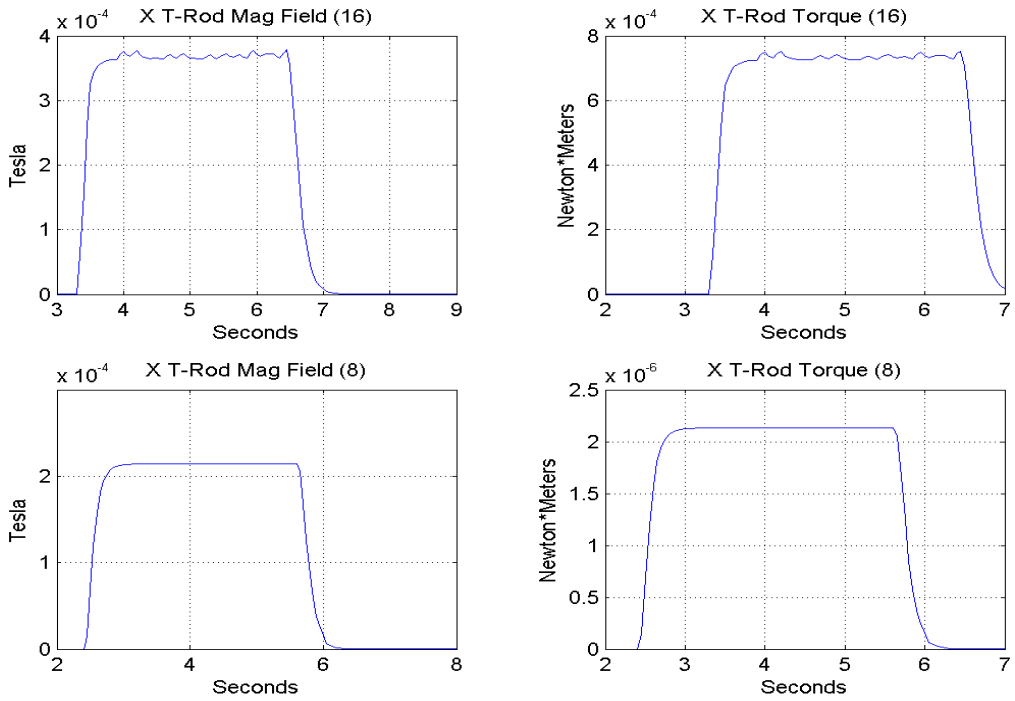


Figure 5.15 NPSAT1 Air–Bearing X Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 16 & 8)

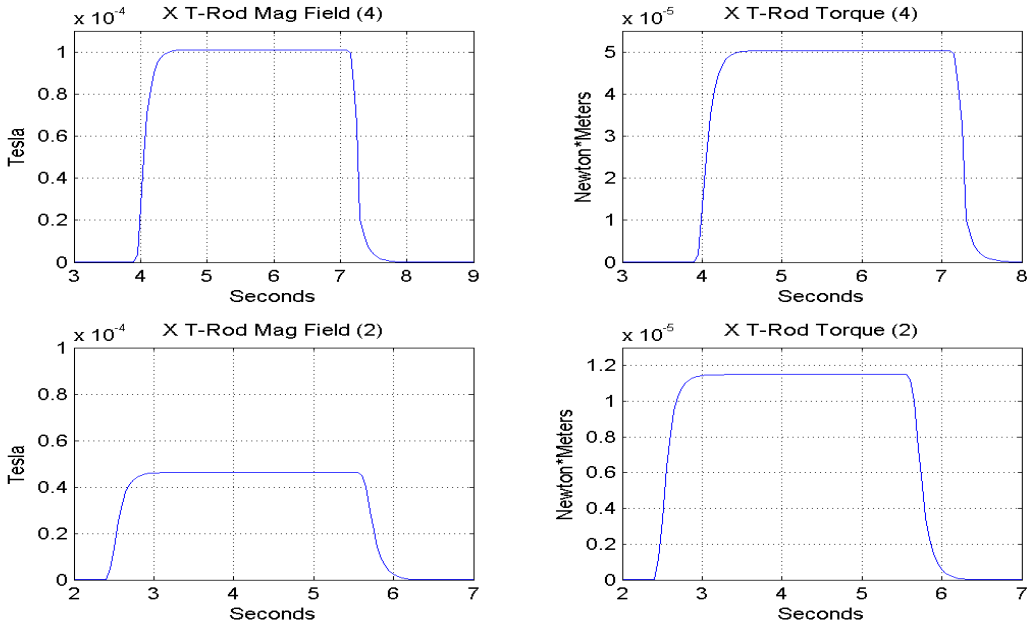


Figure 5.16 NPSAT1 Air–Bearing X Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 4 & 2)

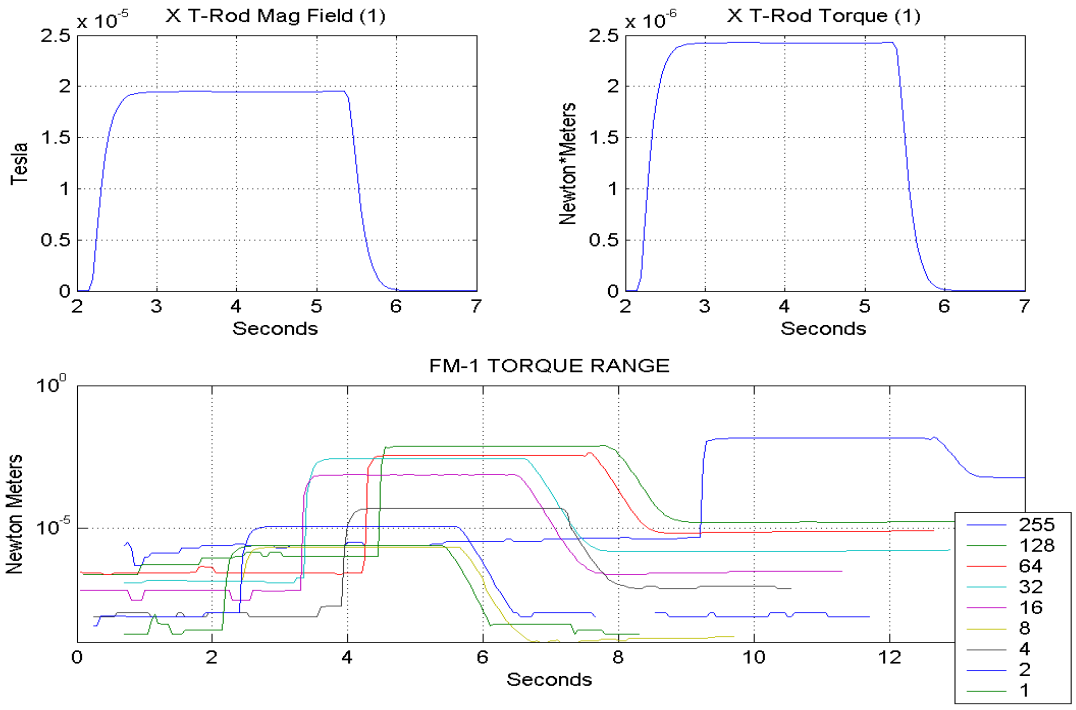


Figure 5.17 NPSAT1 Air–Bearing X Torque Rod Induced Magnetic Field, Associated Torque (Bit Level 1), and Torque Range

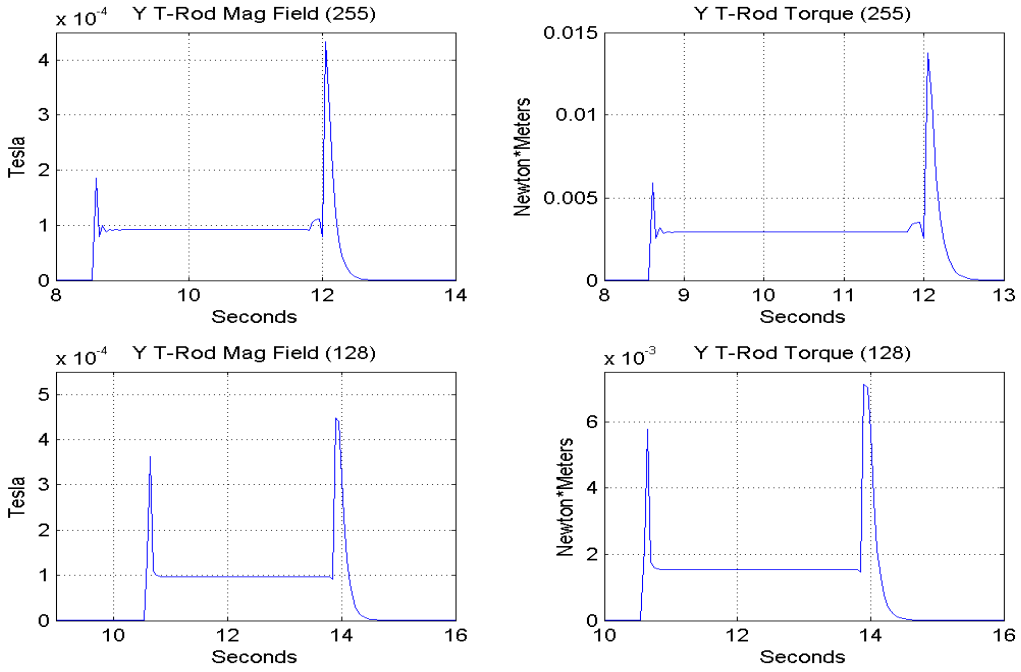


Figure 5.18 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 255 & 128)

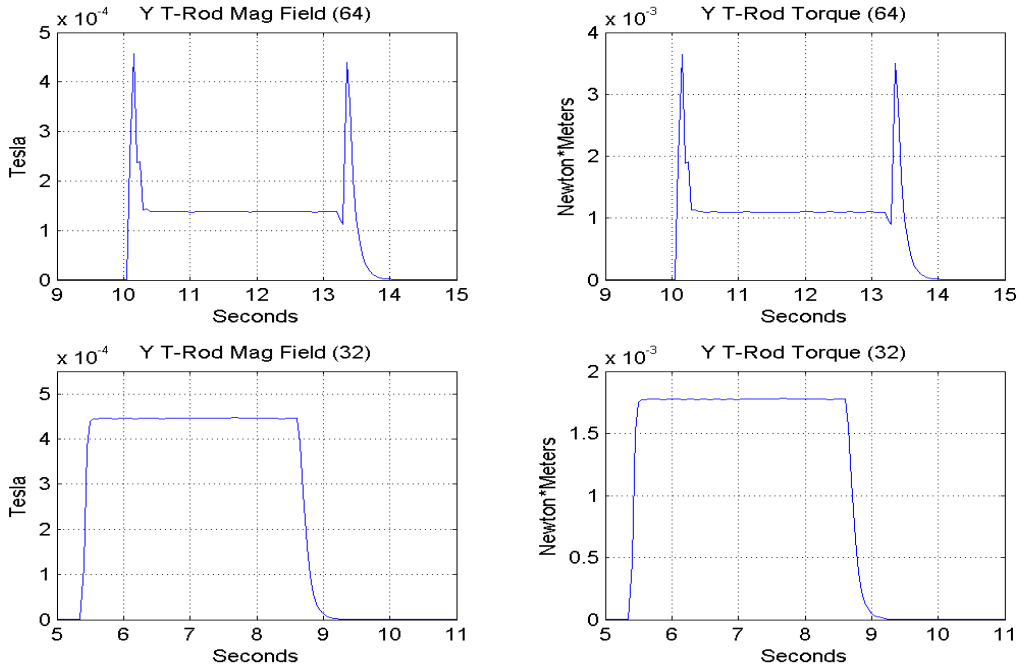


Figure 5.19 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 64 & 32)

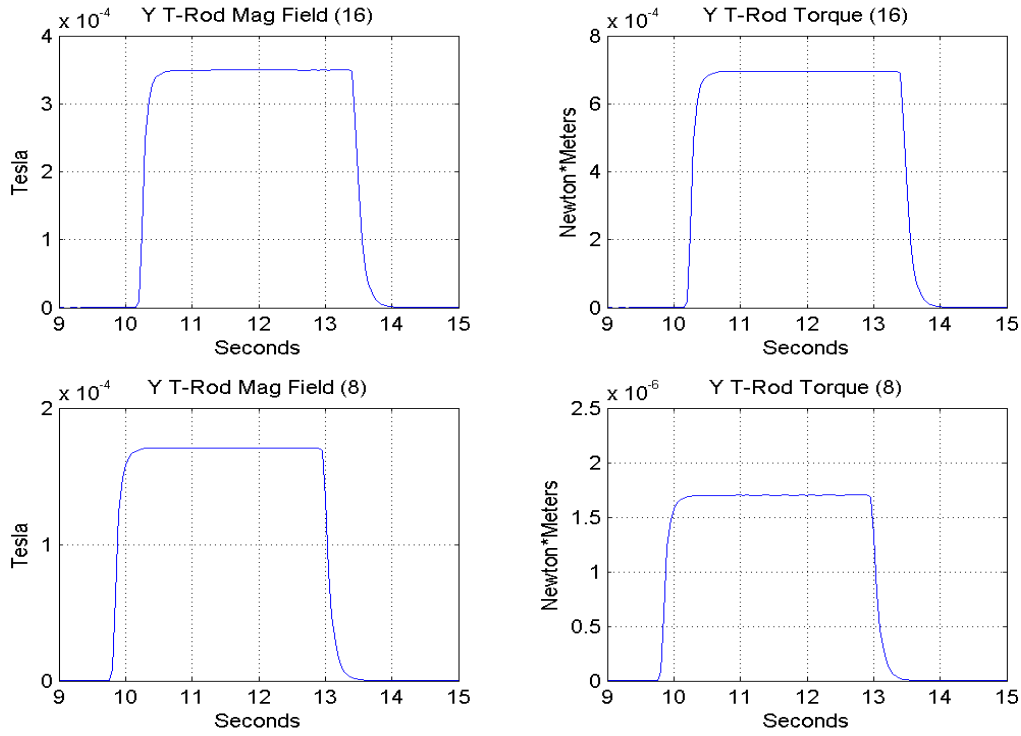


Figure 5.20 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 16 & 8)

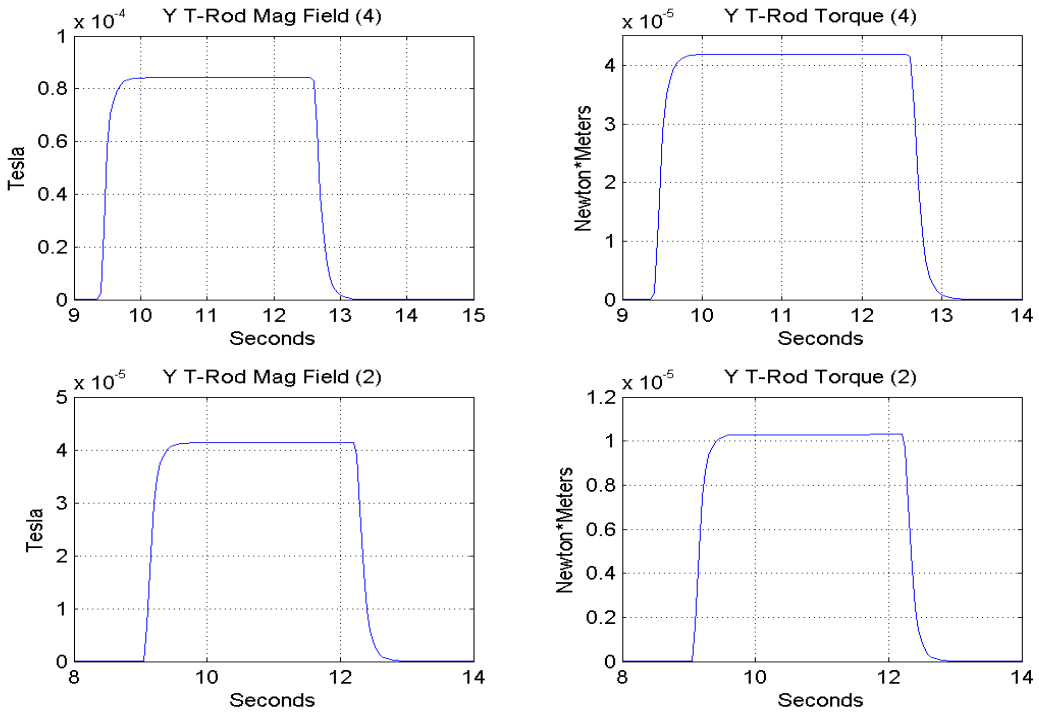


Figure 5.21 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 4 & 2)

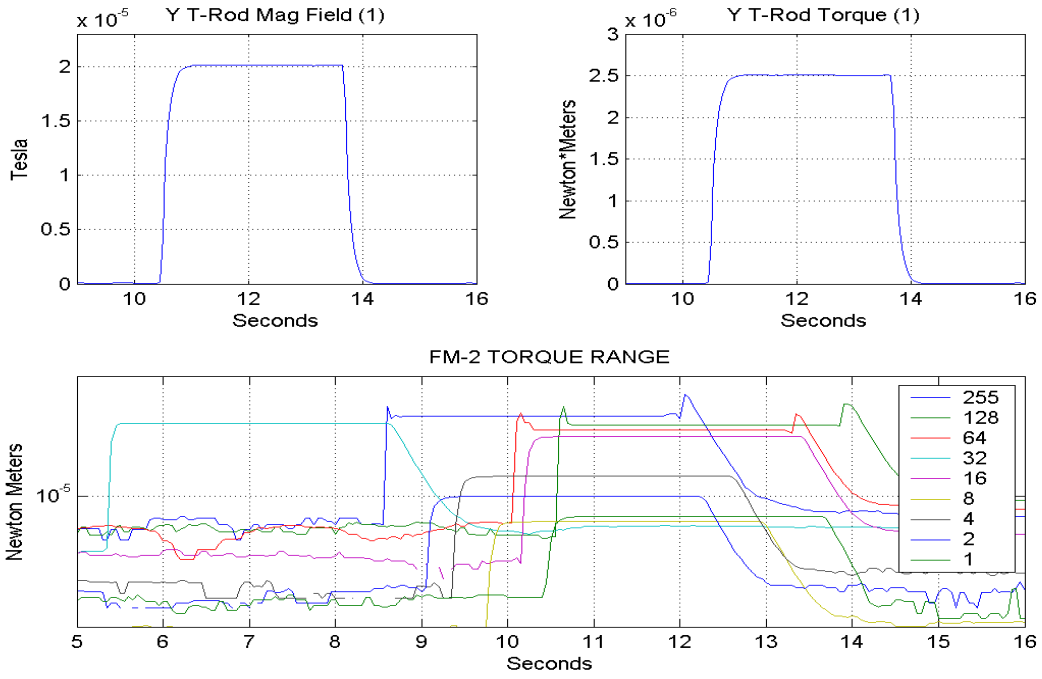


Figure 5.22 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field, Associated Torque (Bit Level 1), and Torque Range

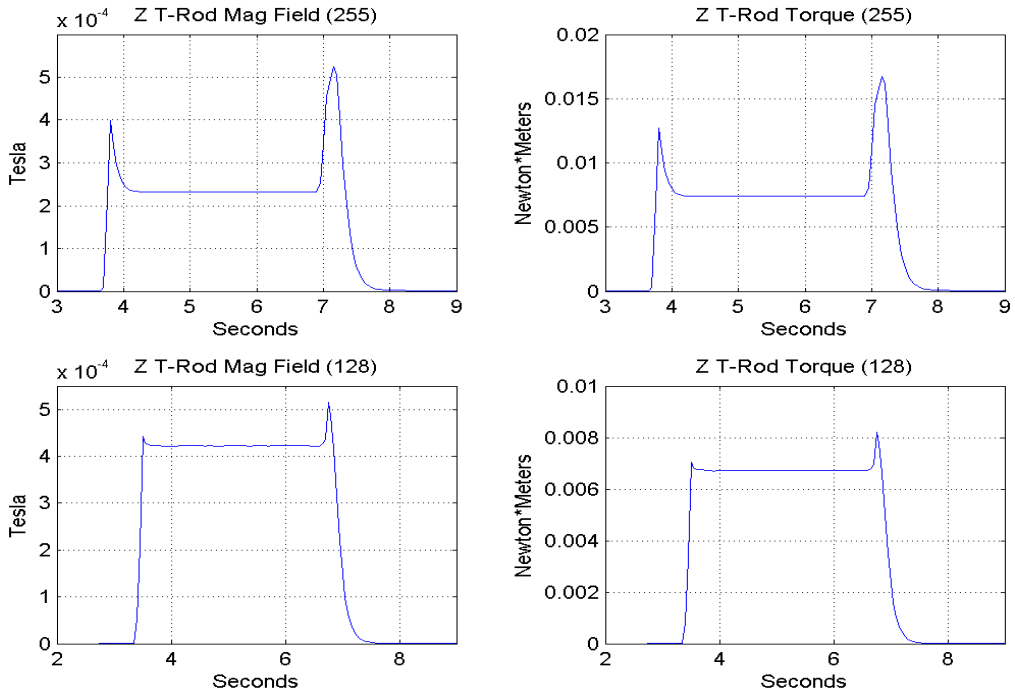


Figure 5.23 NPSAT1 Air–Bearing Z Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 255 & 128)

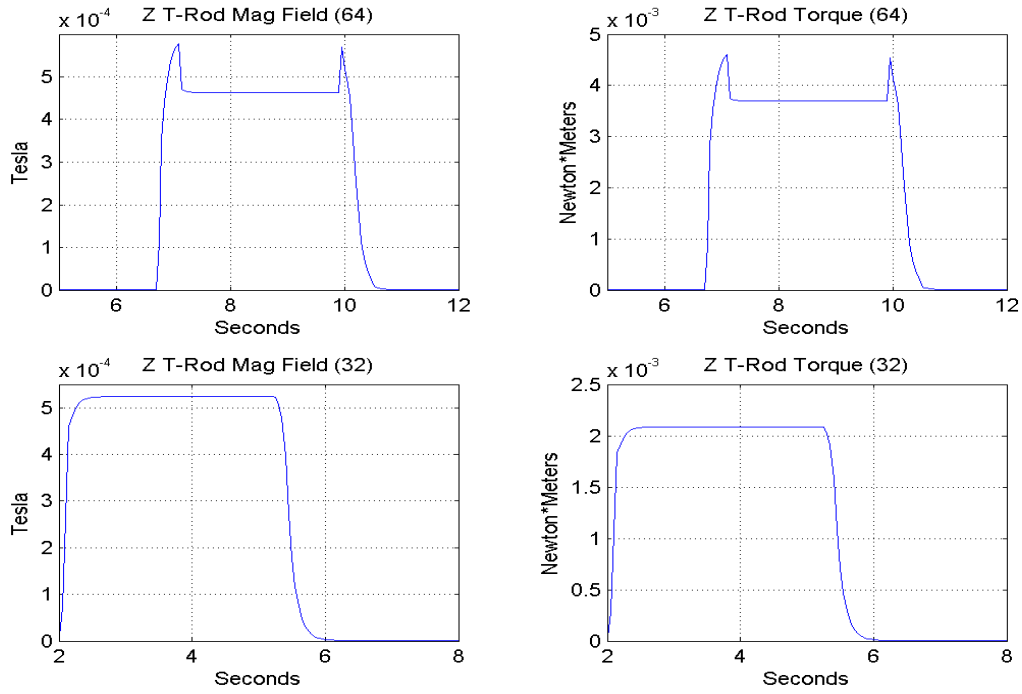


Figure 5.24 NPSAT1 Air–Bearing Z Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 64 & 32)

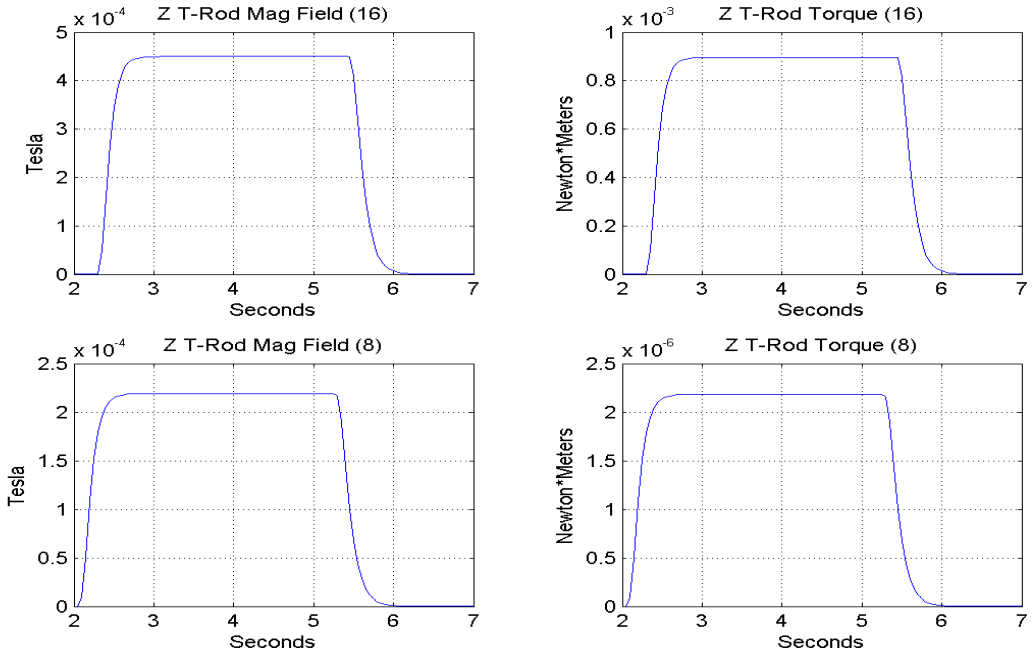


Figure 5.25 NPSAT1 Air–Bearing Z Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 16 & 8)

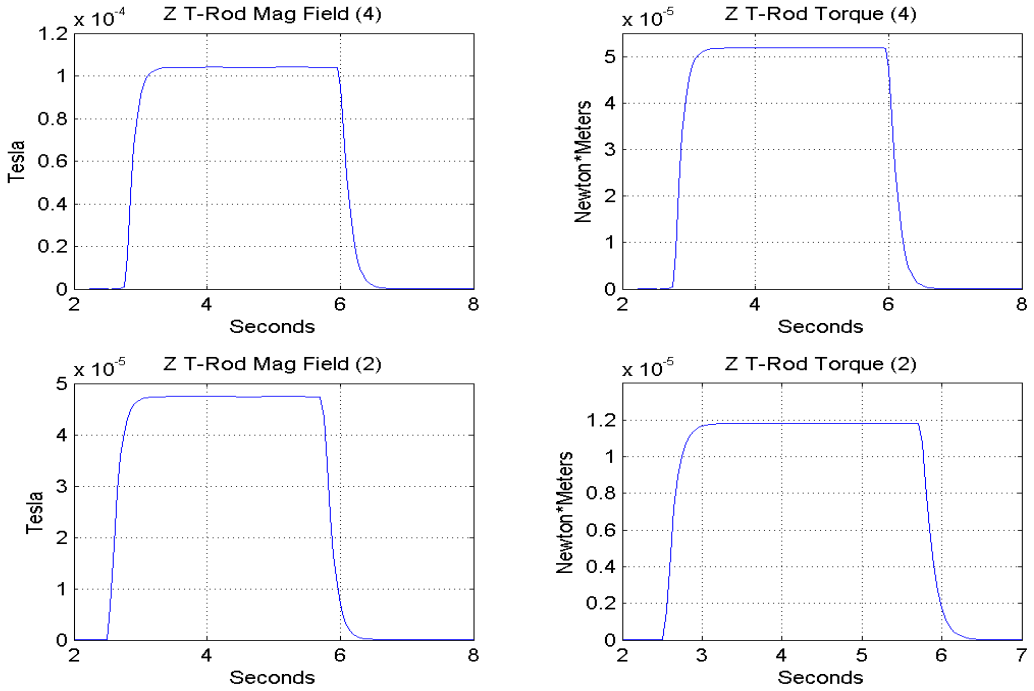


Figure 5.26 NPSAT1 Air–Bearing Z Torque Rod Induced Magnetic Field and Associated Torque (Bit Levels 4 & 2)

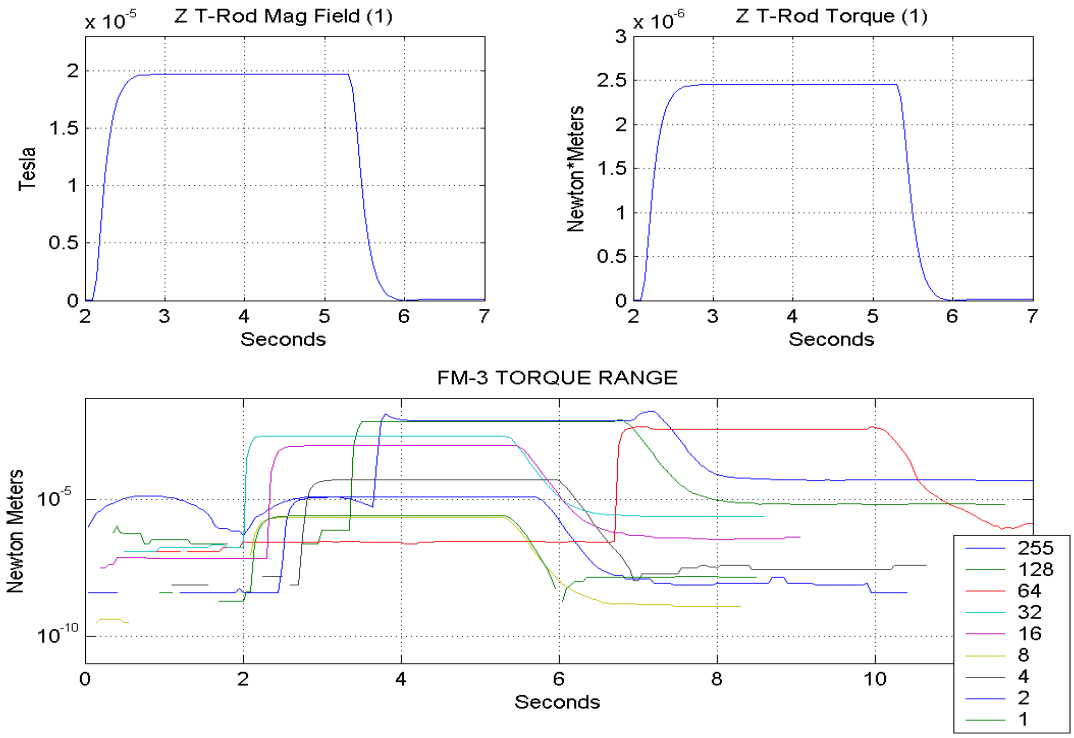


Figure 5.27 NPSAT1 Air–Bearing Y Torque Rod Induced Magnetic Field, Associated Torque (Bit Level 1), and Torque Range

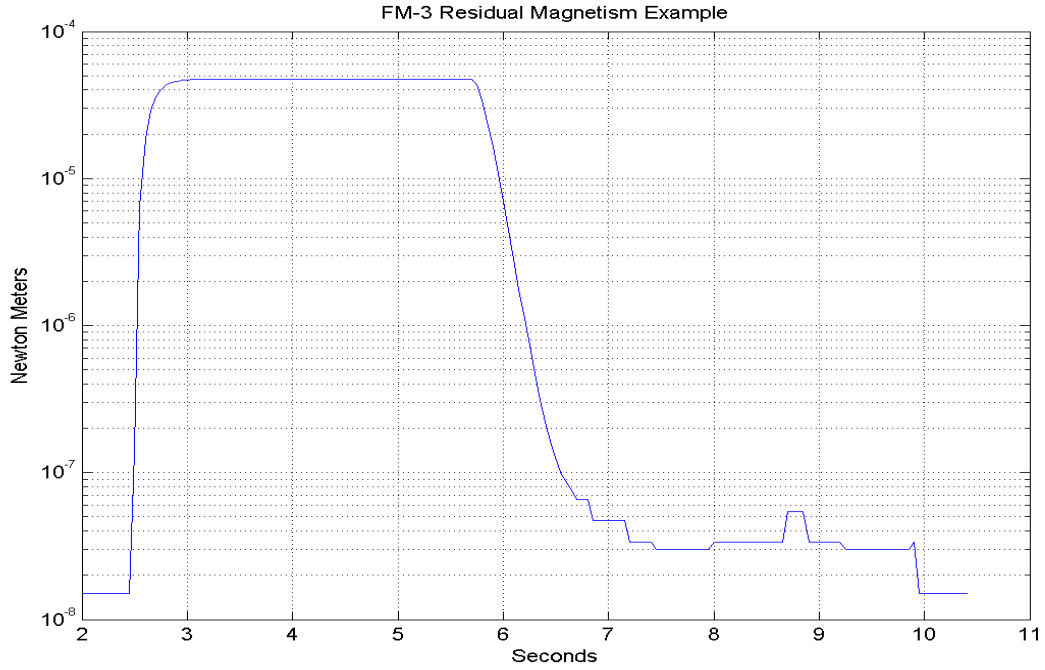


Figure 5.28 NPSAT1 Air–Bearing Y Torque Rod Induced Residual Magnetic Field Sample

Residual magnetism within the spacecraft could possibly upset the stability of the ACS. Possible sources of residual magnetism are the solar array currents, minute yet significant dipoles established in satellite circuitry and hardware, communications equipment, power supplies, and onboard experiments. The determination of the spacecraft residual magnetism was deemed important enough to be classified as an additional research problem. Additionally, once the residual magnetism is quantified, its disturbance can be model as an external disturbance torque. This new disturbance torque can then be accounted for just like the aero, solar, and gravity–gradient disturbance torques were considered.

G. NPSAT1 AIR–BEARING FOLLOW-ON RESEARCH

A goal of this thesis was to prove that the ACS model designed by Leonard was feasible. This was to be accomplished using the air-bearing platform. However, due to thesis completion time constraints, hardware, and software faults, full *hardware-in-the-loop* tests were not fully completed. Achieving this goal was possible; however, further preliminary work needs to be accomplished prior to reinitializing the *hardware-in-the-loop* experiments. Specifically, using the results determined from the input/output current tests, DAC tests, torque driver board tests, and torque rod magnetic field test, it is deemed that more attention needs to be given to the hardware used on the air-bearing platform.

The first step is to initiate a redesign of the torque driver circuit board. This redesign should include a noise reduction analysis that incorporates solid-state components, amplification, and filtering capability. The 40 mV of noise discovered within the circuit boards circuitry is unacceptable.

The second step is to replace the current torque rods with solenoids that are specifically designed and manufactured for the satellite's requirements. The rods used now are commercial solenoids that were purchased prior to a complete understanding of the attitude control system's function.

Once these two procedures are completed, basic *hardware-in-the-loop* examinations could re-commence. Upon completion of the basic *hardware-in-the-loop* tests, the following topics still require exploration installation of motion and recording sensors to the air-bearing platform to capture the ACS fidelity

- installation of Helmholtz Coils that will allow the ACS to be exercised in a magnetic field similar to that in space

- determination of the residual magnetic fields produced by the S/C, onboard equipment, and experiments.

Completion of the follow-on research will allow for the NPSAT1 engineers to fully understand how the attitude control system interacts with external magnetic fields.

In summary, this chapter described the initial *hardware-in-the-loop* tests designed to exercise the NPSAT1 ACS. Specifically, the ACS program was converted from a MATLAB .m file into executable code using Real Time Workshop. Additionally, tests were conducted to determine if all of the air–bearing components functioned as initially designed. Sadly, these tests failed. However, from the failure came useful information that is leading to improvement of follow–on testing and hardware design.

The next chapter presents the conclusion drawn from this thesis research.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

This thesis had three objectives. The first objective was to verify and to validate the NPSAT1 magnetic attitude control system algorithm and simulation model drafted by Leonard [9]. The second objective was to establish a laboratory that would house an air-bearing platform designed to emulate NPSAT1 attitude control system. The final objective was to perform experiments using the air–bearing to exercise the NPSAT1 ACS software and hardware.

Verification and validation of the NPSAT1 magnetic attitude control program and computer simulation model was completed by researching the cited references used by Leonard, drafting a program based on the reference material, and building a model designed to mimic the satellites attitude process. The empirical relationships used in drafting the ACS program and computer model were established in Chapter III. The program and model incorporated these empirical relationships by modeling the external magnetic field the satellite will experience, external disturbance torques, spacecraft dynamics and kinematics, and the control laws required to induce a dipole moment and torque demanded by the ACS.

Comparing the developed models to existing models and known control theory via inspection completed validation of each stage. Specifically, the magnetic field that the spacecraft will experience was compared against two known models. The first was a model developed by the Finnish Meteorlogical Institute [2]. The second was a textual model developed by Wertz in [1]. Both models compared well to Leonard's [9] and to the one drafted for this thesis. ACS model validation was completed by independently constructing the model as Leonard built it. Performing this process allowed for the original model to be dissected and examined against empirical theory discussed in Chapter III. All facets of the model were analyzed and proved to be accurate science.

The second thesis objective was to establish an environment where the ACS program could be tested using hardware. This task built on previous work completed by Schmidt [11] who developed an air–bearing platform. The air-bearing platform was constructed to exhibit similar characteristics that NPSAT1 hold. So, to establish a test bed, a laboratory had to be established. The laboratory was constructed by first establishing a reference grid system that would take into account the air–bearing's body frame, measurement devices frame, and labora-

tory frame. Additionally, the magnetic field within the laboratory was surveyed and mapped. Measuring the individual magnetic field lines and determining the magnitude for a discrete point in the laboratory space completed the mapping. The measured values are important because the values are used by the ACS as an entering argument needed by the ACS program to calculate necessary dipole moment changes.

The magnetic field surveying and mapping progress were completed in three stages. The first survey performed encompassed the entire grid reference system surrounding the air–bearing platform. This initial survey captured a snapshot of what the laboratory's magnetic field resembled and how material in the laboratory affected the field's properties. The second survey narrowed the scope of examination. The second survey concentrated upon the immediate area surrounding the air–bearing. The results of this survey proved that the magnetic field in the air–bearing's immediate vicinity was warped; however, it was mostly static and exhibited very few radical changes in magnitude. The final survey was completed from an air-bearing aspect. Here, the magnetometer used to measure the previous two surveys was mounted to the air–bearing. The air–bearing was rotated 360 degrees about its *Z* axis and inclined by 75 degrees and 105 degrees. Doing so allowed for the survey to capture the magnetic field that would immediately affect the ACS system. Moreover, this survey provided the external magnetic field ephemeris data required by the ACS program.

The third thesis objective was to convert the ACS program from a simulation model into executable code, establish XPC Host and XPC Target communication protocols, and to perform *hardware-in-the-loop* tests that would demonstrate the NPSAT1 ACS functionality.

The first two goals set by the third thesis objective were successfully met. The ACS simulation model was successfully converted from a MATLAB .m file and SIMULINK .mdl file into an ANSI 'C' code executable file. This was accomplished using the SIMULINK Real Time Workshop function. This function proved invaluable because it allows the engineer to model a system and then use that same model as the foundation to perform hardware and software implementation. Additionally, once the ACS model was converted into ANSI 'C' code, the executable code could be immediately loaded into the air–bearing's computer using the SIMULINK XPC Host and XPC Target functions. Doing so allowed the executable model parameters to be adjusted as necessary to ensure processed signals traveled the correct path.

Moreover, the model's executability was successful because the interface between air–bearing's magnetometer, torque rod driver circuit board, micro–controller computer, and host computers was solved through the use of a RS-232 input/output device.

The final goal of the third thesis objective was partially completed since the planned objectives were not entirely accomplished. The objectives were to initiate the ACS program on the air-bearing platform and to witness it control the air–bearing as it would control NPSAT1 in space. However, the *hardware-in-the-loop* tests deviated from the ideal tests planned. The deviations were caused by faults discovered in the micro–controller single board computer, torque rod driver circuit board, and torque rods. Specifically, *hardware-in-the-loop* experiments commenced by attempting to run the ACS program directly after all air–bearing hardware was installed and connected. The tests failed from signals not being received and transmitted correctly. The first impression was that the ACS model was not synchronizing the signals as designed; however, after many tests and reexaminations, focus was shifted to the air–bearing hardware and software. It was discovered that the software written to control the data flow to and from the ACS program was flawed. Additionally, hardware faults were discovered in the torque driver circuit board that prevented the proper digital to analog conversion of torque rod commands.

Following the initial problems discovered, focus was shifted to learning more on how the ACS program, micro–controller, torque rod driver circuit board, and torque rods interacted. This was accomplished by conducting tests between the torque rod driver circuit board and torque rods. Specifically, dipole moment current tests, digital-to-analog signal conversion tests, and torque rod magnetic field tests were conducted. These tests were deemed appropriate since the torque rods are required to operate in extremely low levels of torque fabrication. Test results showed that there was a great deal of noise within the hardware that interfered with the proper current application to the solenoids. Additionally, it was determined that the torque rods currently in use were two large and would not perform to the specifications desired. Therefore, analysis commenced to determine if air coils would be an appropriate substitute.

Though the *hardware-in-the-loop* tests have not been fully completed, the results gained from the tests completed yielded sound scientific information. The data collected now

allows the NPSAT1 Project Team to better understand the magnetic ACS system. Furthermore, the results from the tests identify areas where further research is required.

Specifically, there are four topics of interest that need immediate attention.

The first topic is to continue the preliminary research regarding the appropriate sized torque rods required for NPSAT1 and the air–bearing. This research will include determining the appropriate design and construction of the solenoid. This is important because the existing torque rods were constructed without fully knowing what the ACS program would require.

The second topic of interest is to calculate the residual magnetism that will be present onboard NPSAT1 and the air–bearing. Residual magnetism is produced by any element that can hold a magnetic signature. Therefore, all of NPSAT1's sub–systems, electronics, wiring, and frame magnetism will have to be calculated. The resultant will be considered an external disturbance torque that can be modeled and accounted for when determining the exact amount of torque required by the ACS. If this external disturbance is not accounted for, it could possibly cause the magnetometer to read a skewed magnetic field; thus, causing a miscalculation in the counter torque required by the ACS.

The third topic involves the continuation of the *hardware-in-the-loop* experiments once all of the hardware issues are resolved. This process will continue where this thesis ended.

The final topic of interest will be to advance the air-bearing laboratory. Currently, the air-bearing laboratory is operating at its most basic form. To truly capture the ACS performing its attitude control function, remote sensors, recording devices, and a non–circulating atmosphere are required. The majority of the equipment required has already been acquired; however it has yet to be installed for fully remote operation. Additionally, the desire is to conduct the simulation in an environment that mimics that which will be felt in space. Therefore, the magnetic field within the laboratory will have to be adjusted to resemble that at 600 km above the Earth. Doing so will require the installation of Helmholtz Coils which will allow the magnetic field to be adjusted as desired.

The verification described in the body of this thesis and attention to the above four topics will significantly improve NPSAT1 on–orbit performance.

APPENDIX A. INTERNATIONAL GEOMAGNETIC REFERENCE FIELD–EPOCH 2000

This Appendix contains the International Geomagnetic Reference Field coefficient values and correction factors required to calculate the Earth's magnetic field.

g/h	n	m	1995	2000	SV
g	1	0	-29692	-29619.4	13.3
g	1	1	-1784	-1728.2	11.6
h	1	1	5306	5186.1	-21.2
g	2	0	-2200	-2267.7	-14.4
g	2	1	3070	3068.4	-3.7
h	2	1	-2366	-2481.6	-22.7
g	2	2	1681	1670.9	-3.6
h	2	2	-413	-458	-11.1
g	3	0	1335	1339.6	-1.1
g	3	1	-2267	-2288	-3.5
h	3	1	-262	-227.6	5.6
g	3	2	1249	1252.1	-1.2
h	3	2	302	293.4	-4.5
g	3	3	759	714.5	-8.5
h	3	3	-427	-491.1	-8
g	4	0	940	932.3	-2.7
g	4	1	780	786.8	2.2
h	4	1	262	272.6	1.7
g	4	2	290	250	-8
h	4	2	-236	-231.9	1
g	4	3	-418	-403	4.5
h	4	3	97	119.8	5.1
g	4	4	122	111.3	-1.9
h	4	4	-306	-303.8	-0.3
g	5	0	-214	-218.8	-1.4
g	5	1	352	351.4	0.7
h	5	1	46	43.8	-0.3
g	5	2	235	222.3	-2.6
h	5	2	165	171.9	1.5
g	5	3	-118	-130.4	-1.2
h	5	3	-143	-133.1	2
g	5	4	-166	-168.6	0
h	5	4	-55	-39.3	3.8
g	5	5	-17	-12.9	-0.2
h	5	5	107	106.3	-0.5
g	6	0	68	72.3	0.4
g	6	1	67	68.2	0.3
h	6	1	-17	-17.4	-0.7
g	6	2	68	74.2	0.7
h	6	2	72	63.7	-1.8
g	6	3	-170	-160.9	1.9
h	6	3	67	65.1	-0.2

g	6	4	-1	-5.9	-1.7
h	6	4	-58	-61.2	-0.4
g	6	5	19	16.9	-0.5
h	6	5	1	0.7	-0.2
g	6	6	-93	-90.4	0.7
h	6	6	36	43.8	1.5
g	7	0	77	79	0.2
g	7	1	-72	-74	-0.1
h	7	1	-69	-64.6	0.7
g	7	2	1	0	-0.3
h	7	2	-25	-24.2	0.3
g	7	3	28	33.3	1.1
h	7	3	4	6.2	0.1
g	7	4	5	9.1	0.7
h	7	4	24	24	0.3
g	7	5	4	6.9	0.5
h	7	5	17	14.8	-0.8
g	7	6	8	7.3	-0.3
h	7	6	-24	-25.4	-0.1
g	7	7	-2	-1.2	0.5
h	7	7	-6	-5.8	0.2
g	8	0	25	24.4	0.1
g	8	1	6	6.6	0.2
h	8	1	11	11.9	-0.2
g	8	2	-6	-9.2	-0.5
h	8	2	-21	-21.5	0.1
g	8	3	-9	-7.9	0.2
h	8	3	8	8.5	0.3
g	8	4	-14	-16.6	-0.4
h	8	4	-23	-21.5	0.4
g	8	5	9	9.1	0.2
h	8	5	15	15.5	0.1
g	8	6	6	7	0.5
h	8	6	11	8.9	-0.3
g	8	7	-5	-7.9	-0.7
h	8	7	-16	-14.9	0.4
g	8	8	-7	-7	0.4
h	8	8	-4	-2.1	0.4
g	9	0	4	5	
g	9	1	9	9.4	
h	9	1	-20	-19.7	
g	9	2	3	3	
h	9	2	15	13.4	
g	9	3	-10	-8.4	
h	9	3	12	12.5	
g	9	4	8	6.3	
h	9	4	-6	-6.2	
g	9	5	-8	-8.9	
h	9	5	-8	-8.4	
g	9	6	-1	-1.5	
h	9	6	8	8.4	

g	9	7	10	9.3	
h	9	7	5	3.8	
g	9	8	-2	-4.3	
h	9	8	-8	-8.2	
g	9	9	-8	-8.2	
h	9	9	3	4.8	
g	10	0	-3	-2.6	
g	10	1	-6	-6	
h	10	1	1	1.7	
g	10	2	2	1.7	
h	10	2	0	0	
g	10	3	-4	-3.1	
h	10	3	4	4	
g	10	4	-1	-0.5	
h	10	4	5	4.9	
g	10	5	4	3.7	
h	10	5	-5	-5.9	
g	10	6	2	1	
h	10	6	-1	-1.2	
g	10	7	2	2	
h	10	7	-2	-2.9	
g	10	8	5	4.2	
h	10	8	1	0.2	
g	10	9	1	0.3	
h	10	9	-2	-2.2	
g	10	10	0	-1.1	
h	10	10	-7	-7.4	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. NPSAT1 ACS MODEL H MATLAB PROGRAM

This Appendix contains the MATLAB program written to validate Leonard's work. Specifically, this program models the Earth's magnetic field using the values shown in Appendix A, identifies constants, and calculates the required disturbances as described in Chapter III.

```
% LCDR Eric W. Herbert, United States Naval Postgraduate School
% Spherical Harmonic Magnetic Field Model Calculation
% International Geomagnetic Field 2000-2005 Data resourced from
% 1)  www.ngdc.noaa.gov/IAGA/wg8/igrf.html : downloaded Gaussian
% Coefficients
% 2) Wertz,James,R. "Spacecraft Attitude Determination and Control", 1990
% 3) Leonard, Barry. "NPSAT ADCS Interim Review", August 2001
% 4) www.geo.fmi.fi/MGN/igrf: model used to compare model's results
% IGRF2000 corrected for 2005 based on linear interprolation
% Upon S/C launch, IGRF model will need updating IAW with 2005-2010
% predictions.
% Majority of variable, constants, vectors and matrices were labeled the
% same as Leonard for ease of comparison.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% constants declared
Re=6.37814e6;          % Earth equatorial radius (meters): p. 827 Wertz
h=600e3;               % ideal height above the Earth the satellite is positioned (meters)
r=Re+h;                % total distance s/c is from center of Earth (meters)
we=7.2921e-5;          % Earth spin rate
mu=3.98601e14;         % gravity constant (m^3/s^2)
incln=34.5*pi/180;     % S/C inclination in radians;
beta=15*pi/180;        % solar angle in radians
nu0=-115*pi/180;       % initial subsolar point
alpha0=0*pi/180;       % true anomaly; ideal is 1 or 0
Lg0=0;                 % Greenwich Meridian initial position WRT RAAN; set from ground
                       data
kpre=0;                % nodal precession constant assumed to be zero
wn=kpre*(Re/r)^3.5*cos(incln);   % nodal precession (zero eccentricity)
w0=sqrt(mu/r^3);       % orbital angular velocity (rads/sec)
rho=asin(Re/r);        % Earth's angular radius
V=w0*r;                % linear velocity (m/s)
Cd=2.5;                % drag coefficient ; normally 2<Cd<2.5 used 2.5 worst case SMAD
                       p.366
Kaero=0.5*Cd*V^2;  % Aero torque constant for calculations
psun=4.5567e-6;        % solar pressure constant (N/m^2) ;
Psolar=2*psun;         % sun torque constant SMAD p. 366
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Altitude=linspace(450e3,800e3,8);   % (meters) simulink model lookup table
                                    values
% simulink model look-up table
 Density=[36.1 18 9.25 4.89 2.64 1.47 .837 .439]*1e-13;     % for atmoshpheric denstiy; Wertz
                                                            SMAD
Area=[0.2674 0.2674 .1927]; % projected area in body x,y,z directions (m^2)

dL=[0.002 0.002 0.008];        % predicted center pressure & center of mass delta
                               % in x,y,z directions (m); dL=(cp-cm)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Kinematics and Dynamics: Moment of Inertia Calculations
 %Moment of Inertia Data received from Sakoda 4/20/04

Ixx=5.613560 Iyy=5.625140 Izz=2.561393 Ixy=-0.0017378 Iyz=-0.0259106
Ixz=0.0170608 wrt body axes, SI units kg m^2
%Ixx=5; Ixy=0; Ixz=0; Iyx=0; Iyy=5.1; Iyz=0; Izx=0; Izy=0; Izz=2; % Moments of Inertia (kg
m^2)
Imoi=[Ixx -Ixy -Ixz; -Iyx Iyy -Iyz; -Izx -Izy Izz];     % Moment of Inertia matrix or
                                                       Interita tensor  Sidi p.89

I_inverse=inv(Imoi);                    % inverse Moment of Inertia matrix
a0=Ixx-Iyy+Izz; a1=Iyy-Izz; a2=Ixx-Izz;a3=Iyy-Ixx;          % combination constants:
                                                            Leonard notes pg91
                                    % used for roll, pitch, yaw calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Quaternion equations and angular momentum
phi_0=-0.1; theta_0=0.1; psi_0=-0.1;          % initial Euler angles
                                                (phi,theta,psi(radians)
phidot=-0.1; thetadot=0.1; psidot=0.01;      % initial Euler angle rates (radians/sec)
S1=sin(phi_0/2); S2=sin(theta_0/2); S3=sin(psi_0/2);%
C1=cos(phi_0/2); C2=cos(theta_0/2); C3=cos(psi_0/2);%
q1=S1*C2*C3-C1*S2*S3;               % Euler quaternion elements for 3,2,1 sequence
q2=C1*S2*C3+S1*C2*S3;
q3=C1*C2*S3-S1*S2*C3;
q4=C1*C2*C3+S1*S2*S3;
q=[q1 q2 q3 q4]';                   % Euler quaternion vector; Leonard notes p 27
wx=phidot-psidot*S2-w0*S3*C2;   % angular velocity vector elements Leonard
                                 notes p 40
wy=thetadot*C1+psidot*C2*S3-w0*(C3*C1+S3*S2*S1);
wz=psidot*C2*C1-thetadot*S1-w0*(S3*S2*C1-C3*S1);
w=[wx wy wz]';                      % angular velocity vector Sidi p 89
h0=Imoi*w;                          % angular momentum vector Sidi p 89
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% uncorrected Gaussian Coefficient 'g' and 'h'; units: (nanoTesla):
% n degree range from 1-10  and m order range from 0-10
g_initial=[-29615 -1728 0 0 0 0 0 0 0 0 0;            %2000 IGRF g coef ref (1)
        -2267 3072 1672 0 0 0 0 0 0 0 0;
        1341 -2290 1253 715 0 0 0 0 0 0 0;
        935 787 251 -405 110 0 0 0 0 0 0;
        -217 351 222 -131 -169 -12 0 0 0 0 0;
        72 68 74 -161 -5 17 -91 0 0 0 0;
        79 -74 0 33 9 7 8 -2 0 0 0;
        25 6 -9 -8 -17 9 7 -8 -7 0 0;
        5 9 3 -8 6 -9 -2 9 -4 -8 0;
        -2 -6 2 -3 0 4 1 2 4 0 -1];

g_correct_factor=5*[14.6 10.7 0 0 0 0 0 0 0 0 0;      % 2005 IGRF g coef correction factor
              -12.4 1.1 -1.1 0 0 0 0 0 0 0 0;    % which is multiplied by scaler  5
              0.7 -5.4 0.9 -7.7 0 0 0 0 0 0 0;   % for each year from 2000
              -1.3 1.6 -7.3 2.9 -3.2 0 0 0 0 0 0;% same concept for h correction
              0 -0.7 -2.1 -2.8 -0.8 2.5 0 0 0 0 0;
              1 -0.4 0.9 2 -0.6 -0.3 1.2 0 0 0 0;
              -0.4 -0.4 -0.3 1.1 1.1 -0.2 0.6 -0.9 0 0 0;
              -0.3 0.2 -0.3 0.4 -1 0.3 -0.5 -0.7 -0.4 0 0;
               0 0 0 0 0 0 0 0 0 0 0;
               0 0 0 0 0 0 0 0 0 0 0];

g=g_initial+g_correct_factor;                    %IGRF2000 g coefficient corrected to 2005
h_initial=[0 5186 0 0 0 0 0 0 0 0 0;               % 2000 IGRF h coef
        0 -2478 -458 0 0 0 0 0 0 0 0;
        0 -227 296 -492 0 0 0 0 0 0 0;
        0 272 -232 119 -304 0 0 0 0 0 0;
        0 44 172 -134 -40 107 0 0 0 0 0;
        0 -17 64 65 -61 1 44 0 0 0 0;
        0 -65 -24 6 24 15 -25 -6 0 0 0;
        0 12 -22 8 -21 15 9 -16 -3 0 0;
        0 -20 13 12 -6 -8 9 4 -8 5 0;
        0 1 0 4 5 -6 -1 -3 0 -2 -8];

h_correct_factor=5*[0 -22.5 0 0 0 0 0 0 0 0 0;  % h coefficient correction factor
              0 -20.6 -9.6 0 0 0 0 0 0 0 0;
              0 6 -0.1 -14.2 0 0 0 0 0 0 0;
              0 2.1 1.3 5 0.3 0 0 0 0 0 0;
              0 -0.1 0.6 1.7 1.9 0.1 0 0 0 0 0;
              0 -0.2 -1.4 0 -0.8 0 0.9 0 0 0 0;
              0 1.1 0 0.3 -0.1 -0.6 -0.7 0.2 0 0 0;
              0 0.1 0 0 0.3 0.6 -0.4 0.3 0.7 0 0;
```

```
              0 0 0 0 0 0 0 0 0 0 0;
              0 0 0 0 0 0 0 0 0 0 0];

 h=h_initial+h_correct_factor;     % IGRF2000 h coeffiecient corrected to 2005
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% look up tables for colatitude and longitude increments for calculations
% adapted from Leonard Code
delta_theta=2;                       % colatitude increment (theta) co-latitude increments of 2 de-
grees
theta_min=-90;                       % south pole
theta_max=90;                        % north pole
Jmax=(theta_max-theta_min)/delta_theta;   % max value of J index; 90-(-90)/2=90 values

delta_phi=5;                         % longatude increment allows for complete coverage over entire
Earth circumference
phi_min=-180;                        % west longitude
phi_max=180;                         % east longitude
Imax=(phi_max-phi_min)/delta_phi;% max value of I index ;180-(-180)/5=72 values

for J=1:Jmax+1;
   for I=1:Imax+1;
      theta(J)=(J-1)*delta_theta*pi/180;       % increment theta(co-latitude) values in radians
beginning at the 90deg north
      theta_vector(J)=theta_min+(J-1)*delta_theta;    % theta vector (look up tables for Mag
Field Model)
      phi(I)=(phi_min+(I-1)*delta_phi)*pi/180; %increment phi values in radians
      phi_vector(I)=phi_min+(I-1)*delta_phi;   % phi vector (look up tables for Mag
                                                         Field Model)
   end  % end J loop
end                                  % end I loop

C_theta=cos(theta);                  % sine and cosine of theta; tables
   S_theta=sin(theta);
   if S_theta==0                     % loop to prevent singularity caused by division by zero
       S_theta=eps;
   end
C_phi=cos(phi);                      % sine and cosine of phi
S_phi=sin(phi);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Snm=zeros(10,11);Gnm=zeros(10,11); Pnm=zeros(10,11);            % initialize the Pnm
                                                                 matrix with zeros
part_m=zeros(10,11); part_n=zeros(10,11); part_nm=zeros(10,11);          %initialize
                                                                         partial derivative
                                                                         matrices
% initialization of B field matrix components
```

```
B_r=zeros(J,I); B_theta=zeros(J,I); B_phi=zeros(J,I); B_field=zeros(J,I);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for I=1:Imax+1                    %Loop for longitude; steps from 1 to 73 thus 72 values
   for J=1:Jmax+1                 %Loop for co-latitude; steps from 1 to 91 thus 90 alues
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of Kronecker delta function
      for n=1:10              % 10th degree index
         for m=0:n
            if m==1            % calculation of Kronecker delta
               Kdelta=1;
            else
               Kdelta=0;
            end              % end Kronecker if loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of S ratio
            if m==0
               if n==1
                  Snm(n,m+1)=1;
               else
                  Snm(n,m+1)=Snm(n-1,m+1)*(2*n-1)/n;
               end
            else
               Snm(n,m+1)=Snm(n,m)*sqrt((n-m+1)*(Kdelta+1)/(n+m));
            end                   % end  S ratio if loop


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of Gauss Normalization matrix.
      Gnm(n,m+1)=Snm(n,m+1).*g(n,m+1);
      Hnm(n,m+1)=Snm(n,m+1).*h(n,m+1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Legendre function (K) equation H-9: Wertz p.781
       if n==1          % if n =1 K = 0
          Knm=0;
       else
          Knm(n,m+1)=(((n-1)^2-m^2)/((2*n-1)*(2*n-3)));% for n > 1
       end
             % Leonard USED Km = tril(Km,1) when the matrix evals to this form;
     end               % end of m,n loops
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Recursive Legendre function (P) equation H-8: Wertz p. 781    ok 4/19/04
for n=1:10                        % counter loops on n and m
```

```
    for m=0:n
      if (n-1==0)
          Pm=1;                    % Pk  is the P(n-1,m) term.
          Pn=1;                    % Pn is the P(n-1,n-1) term for the 0,0 case.
       else
          Pm=Pnm(n-1,m+1);   % All other cases
          Pn=Pnm(n-1,(n-1)+1);          % +1 on the indexing to account for starting the
                                        % columns at 1 not 0
       end

       Pk=0;                            % Pk2 is the P(n-2,m) term. Not defined for n<2
       if (n==2) & (m==0)
          Pk=1;                         % For when P(n-2,m) is P(0,0)
       elseif (n>2)
          Pk=Pnm(n-2,m+1);              % All other cases
       end

       if (n==m)
          Pnm(n,m+1)=S_theta(J)*Pn;
       else
          Pnm(n,m+1)=C_theta(J)*Pm-Knm(n,m+1)*Pk;
       end
    end                                 % end recursive P Legendre function for loop
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% recursive partial derivatives calculation Wertz p. 781 equation H-10
%
for n=1:10
    for m=0:n
      if (n-1)==0
          part_m=0;                     % partial derivative (n-1,n-1) term
          part_n=0;                     % partial derivative (n-1,m) case
          Pm=1;
          Pn=1;
      else
          part_m=part_nm(n-1,m+1);
          part_n=part_nm(n-1,(n-1)+1);
          Pm=Pnm(n-1,m+1);
          Pn=Pnm(n-1,(n-1)+1);
      end

      part_k=0;                         % part_k is the partial P(n-2,m) term. Not defined for n<2
      if (n==2) & (m==0)
         part_k=0;                      % when partial P(n-2,m) = (0,0)
      elseif (n>2)
```

```
        part_k=part_nm(n-2,m+1);       % All other cases
      end


      if n==m
        part_nm(n,m+1)=(S_theta(J)*part_n)+(C_theta(J)*Pn);
      else
        part_nm(n,m+1)=(C_theta(J)*part_m)-(S_theta(J)*Pm)-Knm(n,m+1)*part_k;
      end
    end                                 % end partial derivative for loop
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of Single Dipole Magnetic Field Approximation

Kme=(Re/r)^3*sqrt(g(1,1)^2+g(1,2)^2+h(1,2)^2)*1e-9; % dipole approximation (nanoTesla)
sigma=atan(h(1,2)/g(1,2)); %longitude of magnetic dipole (1*e-9 values canx)
epsilon=pi-acos(g(1,1)/sqrt(g(1,1)^2+g(1,2)^2+h(1,2)^2));  % colatitude of magnetic
                                                              dipole (1*e-9 values canx)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% State space model xdot=Ax+Bu+Gw, w = disturbance
% AA3818 Class notes p.66-70
ka=[.02863 .01534 .1309];     % actuator gains; assumed Leonard's values correct
sat=30;                       % saturation; assumed Leonard's values correct
A=[0 0 0 1 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1;
   -4*w0^2*a1/Ixx 0 0 0 0 w0*a0/Ixx;
   0 -3*w0^2*a2/Iyy 0 0 0 0;
   0 0 -w0^2*a3/Izz -w0*a0/Izz 0 0];
B=Kme*[0 0 0;
       0 0 0;
       0 0 0;
       ka(1)/Ixx 0 0;
       0 ka(2)/Iyy 0;
       0 0 ka(3)/Izz];
G=[0 0 0;
   0 0 0;
   0 0 0;
   1/Ixx 0 0;
   0 1/Iyy 0;
   0 0 1/Izz];
Qx=110*eye(6);
Ru=1*[.1 0 0;0 .095 0;0 0 .008];        % LQR weighting matrices; assumed Leonard's
                                        values correct
[K,S,e]=lqr(A,B,Qx,Ru);         % LQR gain calculations
```

```
Aaa=A(1:3,1:3); Aab=A(1:3,4:6); Aba=A(4:6,1:3); Abb=A(4:6,4:6);
% AA3818 notes p 69-70
Ba=B(1:3,1:3); Bb=B(4:6,1:3);
Ka=K(1:3,1:3); Kb=K(1:3,4:6);
Ga=G(1:3,1:3); Gb=G(4:6,1:3);
Lr=0.09*diag([1 .5 .6]);        % Estimator gain by simulation
                                 % Assumed Leonard values correct
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Table inputs for average actuator gains vs. inclination: assumed
% Leonard's values were correct; need to determine if these are the same
% with  my model
Inclination=[0.01 10 20 30 40 50 60 70 80 90 100 110 120]*pi/180;
gx=[.967 .955 .922 .876 .826 .78 .739 .709 .691 .686 .691 .709 .739];
gy=[0.0001 .0678 .256 .46 .632 .762 .857 .923 .965 .981 .965 .923 .857];
gz=[.804 .781 .711 .614 .522 .446 .39 .353 .335 .333 .335 .353 .39];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of B field components (unit: Wb*m)
% Magnetic Field calculation for geocentric distance
for n=1:10
   Br_innersum=0;        % initialize sum for geocentric distance
   Btheta_innersum=0;  % initialize sum for coelevation mag field measurement
   Bphi_innersum=0;     % initialize sum for longitude magnetic field measurement

% partial calculation of B field components using Wertz equation H-12,
 for m=0:n

Br_innersum=Br_innersum+(Gnm(n,m+1)*cos(m*phi(I))+Hnm(n,m+1)*sin(m*phi(I)))*Pnm(
n,m+1);

Btheta_innersum=Btheta_innersum+(Gnm(n,m+1)*cos(m*phi(I))+Hnm(n,m+1)*sin(m*phi(I))
)*part_nm(n,m+1);
      Bphi_innersum=Bphi_innersum+(m*(-
Gnm(n,m+1)*sin(m*phi(I))+Hnm(n,m+1)*cos(m*phi(I))))*Pnm(n,m+1);
   end

% B field components summation using Wertz equation H-12
   B_r(J,I)=B_r(J,I)+((Re/r)^(n+2))*(n+1)*Br_innersum;         % radial component
                                                                 (outward positive)
   B_theta(J,I)=B_theta(J,I)+((Re/r)^(n+2))*Btheta_innersum; % coelevation
                                                                component (south positive)
   B_phi(J,I)=B_phi(J,I)+((Re/r)^(n+2))*Bphi_innersum;   % azimuthal or
                                                                longitudnal component (east posi-
                                                               tive)
end
```

```
   B_theta(J,I)=-B_theta(J,I);                         % gain calculations
   B_phi(J,I)=(-1/S_theta(J))*B_phi(J,I);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end                              % ending co-lat/long for J & I loops
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculation of Earth's Magnetic Field based upon Br, Btheta, Bphi

B_field=sqrt(B_r.^2+B_theta.^2+B_phi.^2);
% end Earth magnetic model calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format short g
V=20000:2000:50000;
figure(1)          % plot Earth's Magnetic Field Model using theta, phi and B_field
c=contour(phi_vector,fliplr(theta_vector),B_field,V);grid on

clabel(c);
title('2005 IGRF Magnetic Field Total Intensity at 600 KM Above the Earth (2000nT Incre-
ments)') ;
xlabel ('Longitude in Degrees');
ylabel ('Latitude in Degrees');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%sim('acs_eric')
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. NPSAT1 ACS MODEL L MATLAB PROGRAM [FROM REF. 9.]

This Appendix contains the original NPSAT1 attitude control system MATLAB program written by Leonard. This program was verified and validated by the program in Appendix B.

```
%NPSAT1ACSData          (pages C1-C4)
clear
tic
Re=6371.2e3;mu=398601.2e9;we=7.2921e-5;%earth radius, gravity and spin rate
Altitude=[450 500  550      600   650   700  750    800]*1e3;%look up table data for
                                                                    aero
Density=[36.1 18    9.25    4.89  2.64  1.47 .837  .439]*1e-13;%max density table data
%Density=[11.3 4.89 2.21 1.04  .515 .272  .155 .0963]*1e-13;%mean density table
                                                                    data

h=600e3;incln=(35)*pi/180;beta=15*pi/180;%altitude,inclination, solar angle
nuo=-115*pi/180;alphao=0*pi/180;%initial subsolar point and true anomaly
Lgo=0;%initial position of Greenwich Meridian WRT RAAN
kpre=-9.9639/24/3600/180*pi; %nodal precession constant assumed to be zero here
wn=kpre*(Re/(Re+h))^3.5*cos(incln);%nodal precession (zero eccentricity)
wo=sqrt(mu/(Re+h)^3);V=wo*(Re+h); %orbital angular and linear velocity
rho=asin(Re/(h+Re)); %earth angular radius

Cd=2.5;psun=4.5e-6;%drag coefficient and solar pressure constant~N/m^2
Kaero=0.5*Cd*V^2
Psolar=2*psun; % constants for aero and solar torque calc.

Area=[0.2674 0.2674 .1927]; %projected area~m^2 in body x,y,z directions
dL=[.002 .002 .008]; %predicted (cp-cm)~m  in body x,y,z directions

Ix=5;Iy=5.1;Iz=2;Ixy=0;Ixz=0;Iyz=0; % moments of inertia (MOI)~kg.m^2
Imoi=[Ix -Ixy -Ixz;-Ixy Iy -Iyz;-Ixz -Iyz Iz]; %MOI matrix
Iinv=inv(Imoi); %MOI matrix inverse
ao=Ix-Iy+Iz;a1=Iy-Iz;a2=Ix-Iz;a3=Iy-Ix;%MOI combination constants

pho=-.1;tho=.1;pso=-.1; %initial Euler angles (phi,theta,psi) ~r
phdo=-0.1;thdo=0.1;psdo=0.01;  % initial Euler angle rates ~r/s
%phdo=0;thdo=0;psdo=0;  % initial Euler angle rates ~r/s

%calculation of initial quaternion (qo) and angular momentum (Ho):
```

```
  s1=sin(pho/2);s2=sin(tho/2);s3=sin(pso/2);c1=cos(pho/2);c2=cos(tho/2);
  c3=cos(pso/2); q1o=s1*c2*c3-c1*s2*s3;q2o=c1*s2*c3+s1*c2*s3;%Wie pg.321
  q3o=c1*c2*s3-s1*s2*c3;q4o=c1*c2*c3+s1*s2*s3;
  S1=sin(pho);S2=sin(tho);S3=sin(pso);C1=cos(pho);C2=cos(tho);C3=cos(pso);
  wxo=phdo-psdo*S2-wo*S3*C2;
  wyo=thdo*C1+psdo*C2*S1-wo*(C3*C1+S3*S2*S1);
  wzo=psdo*C2*C1-thdo*S1-wo*(S3*S2*C1-C3*S1);
qo=[q1o q2o q3o q4o];
Ho=Imoi*[wxo wyo wzo]';

%Calculation of spherical harmonic magnetic field model (Wertz pp779-783)
r=Re+h;a=Re; %definitions to match Wertz pg 780

% IGRF Epoch 2000 Guassian coefficients ~ n.T    :
gg=[-29615 -1728 0 0 0 0 0 0 0;-2267 3072 1672 0 0 0 0 0 0;...
     1341 -2290 1253 715 0 0 0 0 0;935 787 251 -405 110 0 0 0 0;...
     -217 351 222 -131 -169 -12 0 0 0;72 68 74 -161 -5 17 -91 0 0;...
     79 -74 0 33 9 7 8 -2 0;25 6 -9 -8 -17 9 7 -8 -7];
hh=[0 5186 0 0 0 0 0 0 0;0 -2478 -458 0 0 0 0 0 0;...
      0 -227 296 -492 0 0 0 0 0;0 272 -232 119 -304 0 0 0 0;...
      0 44 172 -134 -40 107 0 0 0;0 -17 64 65 -61 1 44 0 0;...
      0 -65 -24 6 24 15 -25 -6 0;0 12 -22 8 -21 15 9 -16 -3];
Kme=(a/r)^3*sqrt(gg(1,1)^2+gg(1,2)^2+hh(1,2)^2)*1e-9;%dipole strength~Tesla

ka=[.02863  .01534  .1309];sat=30;%actuator gains and saturation
%ka=[.03  .02  .12]; sat=30;
%Calc of state space A,B,G matrices follows (xdot=Ax+Bu+Gw,w=disturbance)
A=[0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1;-4*wo^2*a1/Ix 0 0 0 0 wo*ao/Ix;...
   0 -3*wo^2*a2/Iy 0 0 0 0;0 0 -wo^2*a3/Iz -wo*ao/Iz 0 0];
B=Kme*[0 0 0;0 0 0;0 0 0;ka(1)/Ix 0 0;0 ka(2)/Iy 0;0 0 ka(3)/Iz];
G=[0 0 0;0 0 0;0 0 0;1/Ix 0 0;0 1/Iy 0;0 0 1/Iz];
C=[eye(3),zeros(3)];


Qx=110*eye(6); Ru=1*[.1 0 0;0 .095 0;0 0 .008]; %LQR weighting matrices
%Qx=110*eye(6); Ru=.1*[.1 0 0;0 .095 0;0 0 .008]; %LQR weighting matrices
[K,S,e]=lqr(A,B,Qx,Ru); %LQR gain calculation

%partitioning of A,B,G matrices required for reduced order estimator:
Aaa=A(1:3,1:3);Aab=A(1:3,4:6);Aba=A(4:6,1:3);Abb=A(4:6,4:6);
Ba=B(1:3,1:3);Bb=B(4:6,1:3);Ka=K(1:3,1:3);Kb=K(1:3,4:6);
Ga=G(1:3,1:3);Gb=G(4:6,1:3);
Lr=0.09*diag([1 .5 .6]); %Estimator gain found by simulation
%Lr=0.08*.1*diag([1 1 .5])
```

```
del_t=5;tmin=0;tmax=180;%colatitude increment, min, max (theta)
Jmax=(tmax-tmin)/del_t; % max value of J index
del_p=10;pmin=-180;pmax=180; %longitude increment, min, max (phi)
Imax=(pmax-pmin)/del_p;  %max value of I index

for J=1:Jmax+1;
  t=(J-1)*del_t*pi/180;Ct=cos(t);St=sin(t); %sine & cosine of theta
  th(J)=(J-1)*del_t; %theta vector (look up tables Mag Fld Model)
for I=1:Imax+1;
  p=(pmin+(I-1)*del_p)*pi/180;Cp=cos(p);Sp=sin(p);%sin & cos phi
  phi(I)=pmin+(I-1)*del_p;%phi vector (look up tables Mag Fld Model)
 % Calculation of Legendre function constant (Km) follows
  for i=1:8
    for j=1:i+1
      Km(i,j)=((i-1)^2-(j-1)^2)/(2*i-1)/(2*i-3);
      if i==1; Km=0;end
    end
  end
 Km=tril(Km,1);

 %Calculation of Legendre function (P) follows
  for i=1:8
    for j=1:i+1
      P(1,1)=Ct;
      if i==1 & j==2;P(i,j)=St*1;end
      if i>1 & j==i+1;P(i,j)=St*P(i-1,j-1);end

      if i==2;Pk=1;end;
      if i>2;Pk=P(i-2,j);end
      if i>1 & j~=i+1;P(i,j)=Ct*P(i-1,j)-Km(i,j)*Pk;end
    end
  end
 P=tril(P,1);

 % Calculation of partial of Legendre function WRT theta (dp) follows
  for i=1:8
    for j=1:i+1
      dp(1,1)=-St;
      if i==1 & j==2;dp(i,j)=St*0+Ct*1;end
      if i>1 & j==i+1;dp(i,j)=St*dp(i-1,j-1)+Ct*P(i-1,j-1);end
      if i==2;dpk=0;end;
      if i>2;dpk=dp(i-2,j);end
      if i>1 & j~=i+1;dp(i,j)=Ct*dp(i-1,j)-St*P(i-1,j)-Km(i,j)*dpk;end
    end
   end
```

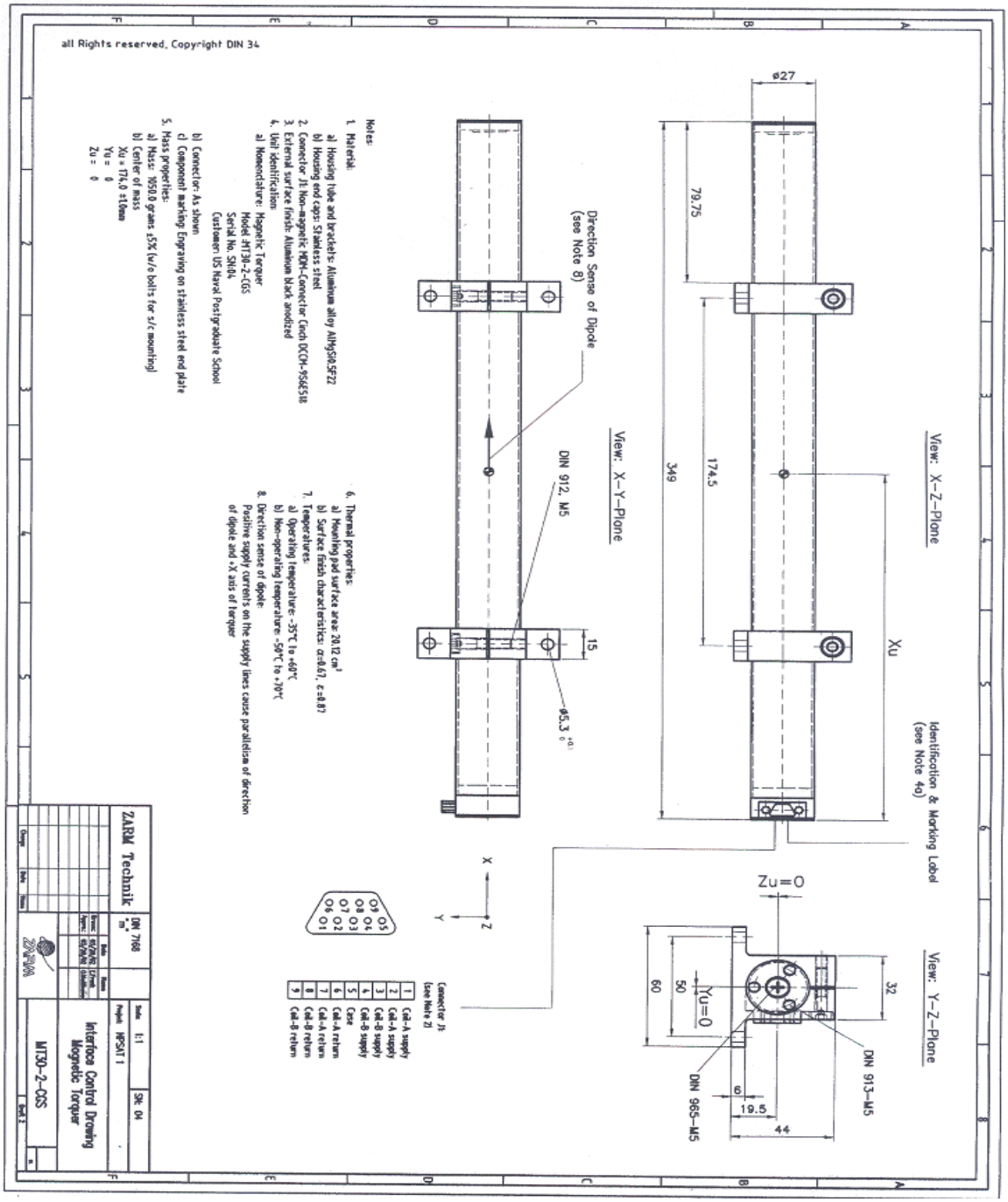
```
 dp=tril(dp,1);

 % Calculation of Gausian coefficient norm. factor (S) follows
 for i=1:8
   for j=1:9
     dm=0;if j==2;dm=1;end
     if j==1 & i==1; S=1;end
     if j==1 & i~=1;S(i,j)=S(i-1,j)*(2*i-1)/i;end
     if j>1;S(i,j)=S(i,j-1)*sqrt((i-(j-1)+1)*(dm+1)/(i+(j-1)));end
   end
 end
S=tril(S,1);

%Calculation of normalized Gaussian coefficients ~ n.T  :
Gg=S.*gg; Hh=S.*hh;

 for i=1:8
   j=1:i+1;m=j-1;
   F(i,j)=Gg(i,j).*cos(m*p)+Hh(i,j).*sin(m*p); %definition
 end
 for i=1:8
   j=1:i+1;m=j-1;
   f(i,j)=m.*(-Gg(i,j).*sin(m*p)+Hh(i,j).*cos(m*p));% definition
 end

 if St==0; St=0.001; ;end %singularity avoidance in following loop
 %Calculation of altitude dependent coefficients follows
 for i=1:8
  aa1(i)=(i+1)*(a/r)^(i+2); aa2(i)=-aa1(i)/(i+1); aa3(i)=aa2(i)/St;
 end

%Calculation of radial,theta and phi magnetic field components~Tesla
FPS=sum((F.*P)')'  ;   B1=aa1'.*FPS;   Br8=sum(B1)*1e-9;
FdpS=sum((F.*dp)')';   B2=aa2'.*FdpS;  Bt8=sum(B2)*1e-9;
fPS=sum((f.*P)')'  ;   B3=aa3'.*fPS;   Bp8=sum(B3)*1e-9;

%Calculation of normalized field commponents (input to lookup tables)
br8(I,J)=Br8/Kme;
bt8(I,J)=Bt8/Kme;
bp8(I,J)=Bp8/Kme;

%Calculation of field vector normalized magnitude:
b8(I,J)=(Br8.^2+Bt8.^2+Bp8.^2).^0.5/Kme;

end  %end I loop
```

```
end  %end J loop

% Table inputs for average actuator gains vs inclination follows:
Inclination=[0.01 10 20 30 40 50 60 70 80 90 100 110 120]*pi/180;
gx=[.967 .955 .922 .876 .826 .78 .739 .709 .691 .686 .691 .709 .739];
gy=[0.0001 .0678 .256 .46 .632 .762 .857 .923 .965 .981 .965 .923 .857];
gz=[.804 .781 .711 .614 .522 .446 .39 .353 .335 .333 .335 .353 .39];

format short g
V=20000:2000:50000;
figure(1)              % plot Earth's Magnetic Field Model using theta, phi and B_field
c=contour(phi,th,b8,V);grid on

% clabel(c);
% title('2005 IGRF Magnetic Field Total Intensity at 600 KM Above the Earth (2000nT In-
crements)') ;
% xlabel ('Longitude in Degrees');
% ylabel ('Latitude in Degrees');



toc
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. NPSAT1 SOLENOID CROSS–SECTIONAL VIEW [FROM REF. 8.]

This Appendix displays the cross–sectional view of the MICROCOSM magnetic torque rods tested by this thesis.



THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. NPSAT1 ACS SIMULINK MODEL H

This Appendix displays the overall and sub-models developed in SIMULINK to verify Leonard's ACS model and control program. Sub–models shown include the spacecraft external disturbance torques, kinematics, dynamics, and ACS control laws.

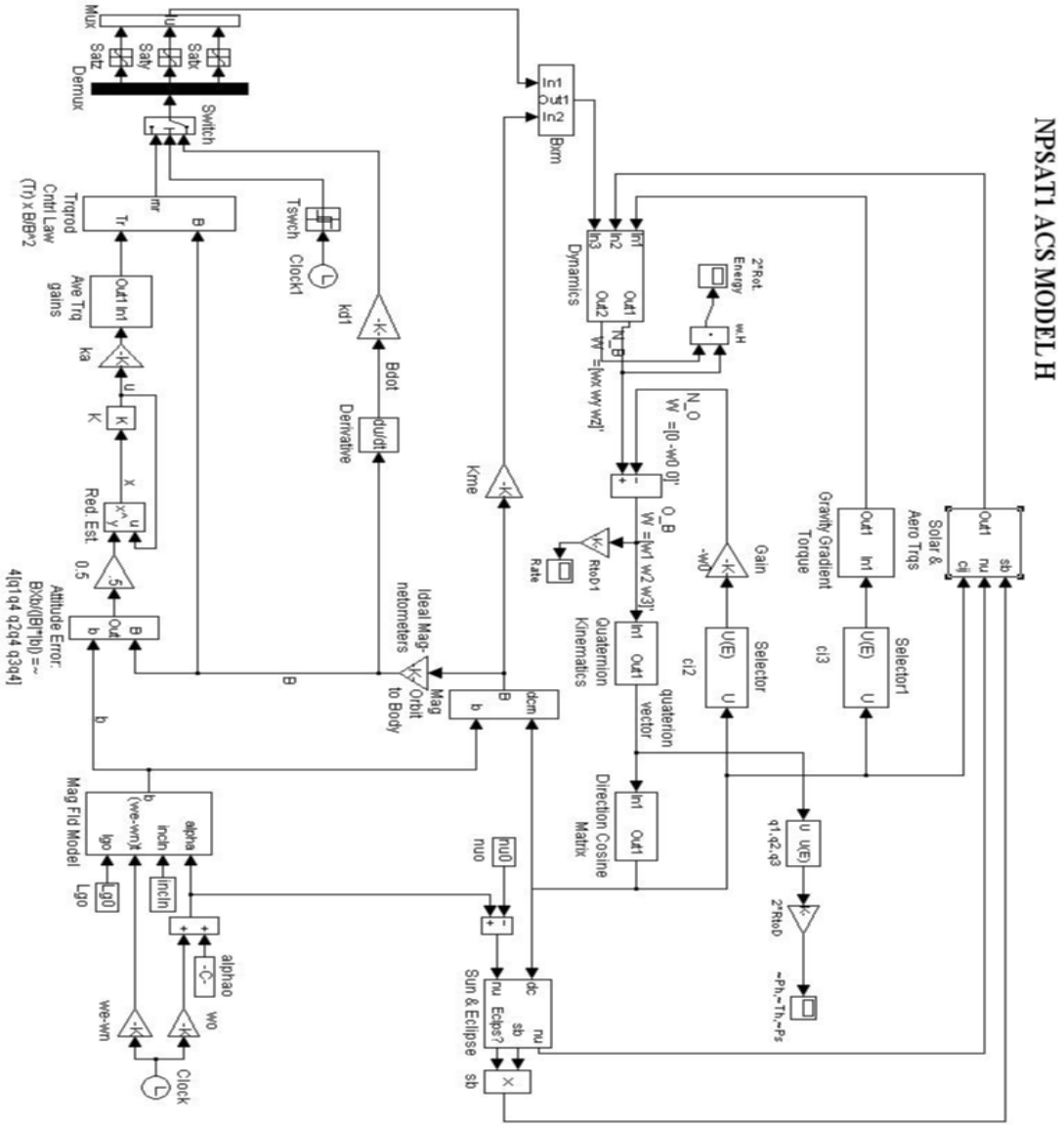


Figure E.1 NPSAT1 Magnetic Attitude Control System SIMULINK Model [After Ref. 9.]

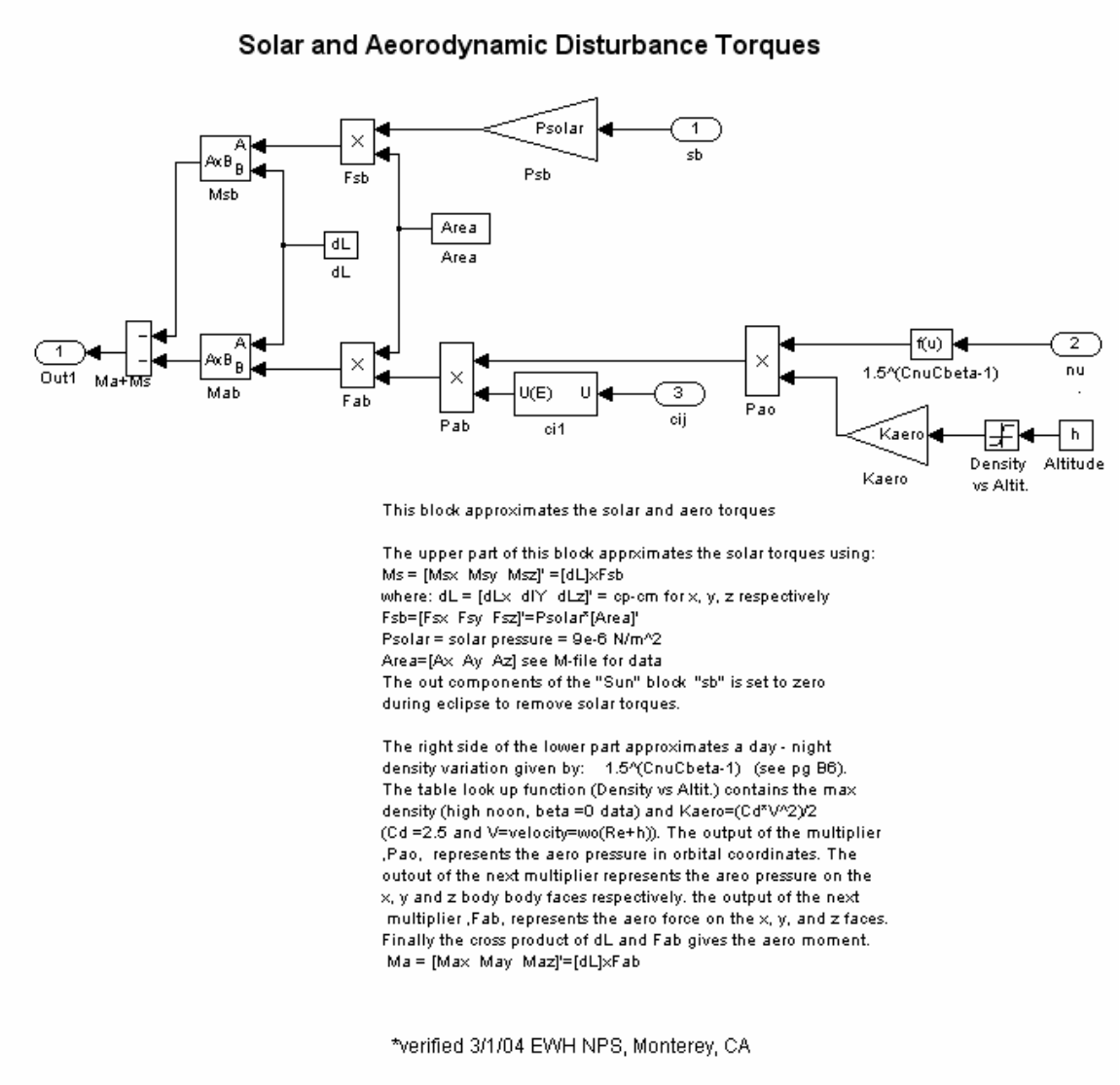


Figure E.2 NPSAT1 Solar and Aero Disturbance Torque Sub–Model [After Ref. 9.]

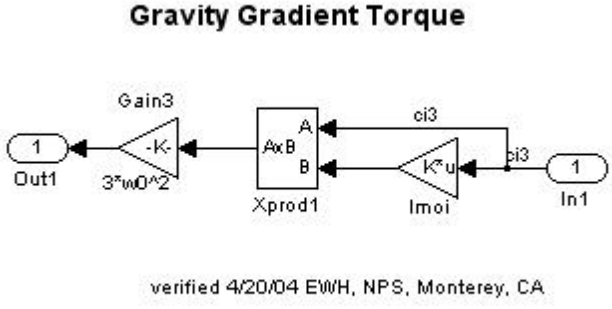


Figure E.3 NPSAT1 Gravity–Gradient Disturbance Torque Sub–Model [After Ref. 9.]

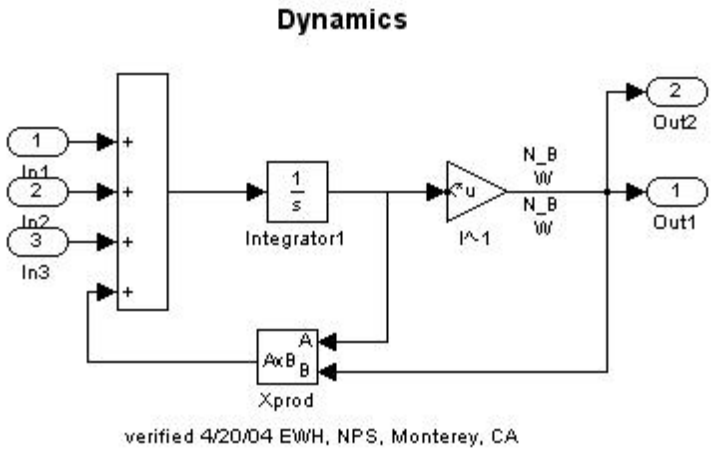


Figure E.4 NPSAT1 Dynamics Sub–Model [After Ref. 9.]

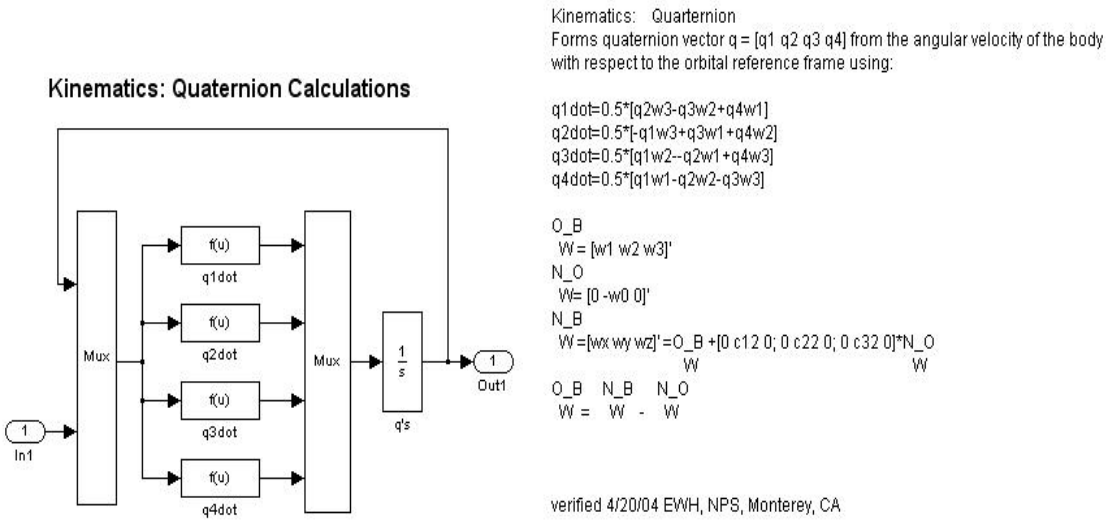


Figure E.5 NPSAT1 Kinematics Sub–Model [After Ref. 9.]

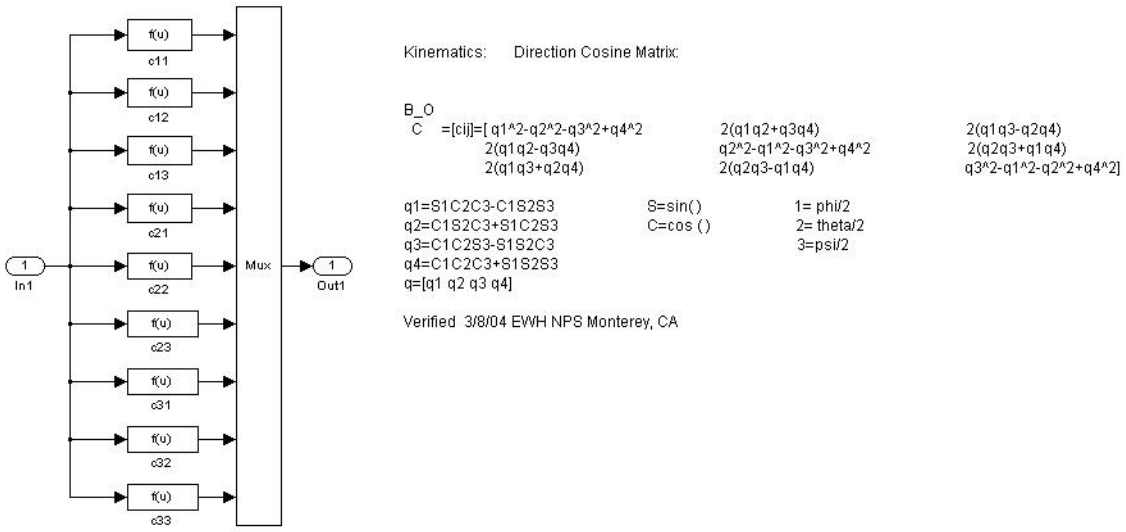
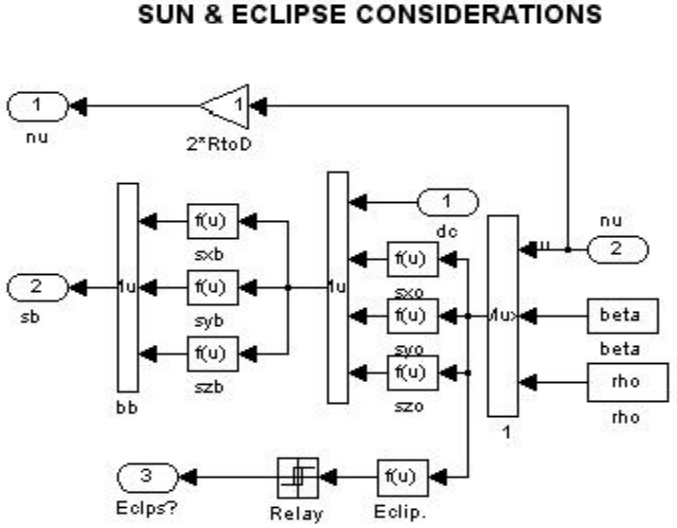


Figure E.6 NPSAT1 Kinematics–Direction Cosine Matrix Sub–Model [After Ref. 9.]



The right side of this block transforms the sun vector into orbital coordinates using:
sxo = -Cbeta*Snu
syo = -Sbeta
szo = -Cbeta*Cnu
The left side of this transforms the sun vector in orbital coordinates into body coordinates using:
sbx = c11sxo+c12syo+c13szo
sby = c21sxo+c22syo+c23szo
sbz = c31sxo+c32syo+c33szo
The relay in the lower part of this block outputs a "1" in sunlight and a "0" during eclipse
the relay is driven by the function Cnu+(Cbeta) /Crho
where: beta = minimum angle between the sun vector and the orbit plane on a given day
nu = orbit angle between the satellite and the subsolar point
rho = apparent angular radius of the earth
C = cosine and S = sine

verified 3/1/04 EWH NPS Monterey, CA

Figure E.7 NPSAT1 Solar Eclipse Sub–Model [After Ref. 9.]

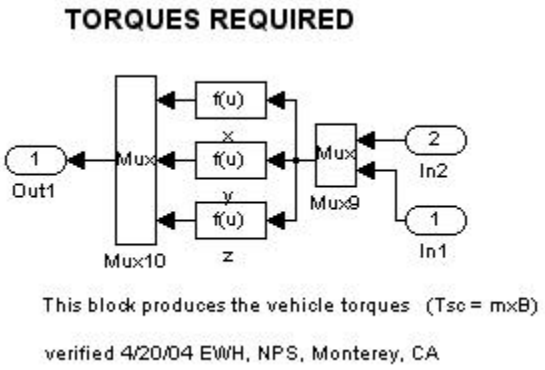


Figure E.8 NPSAT1 Ordered Torque Sub–Model [After Ref. 9.]

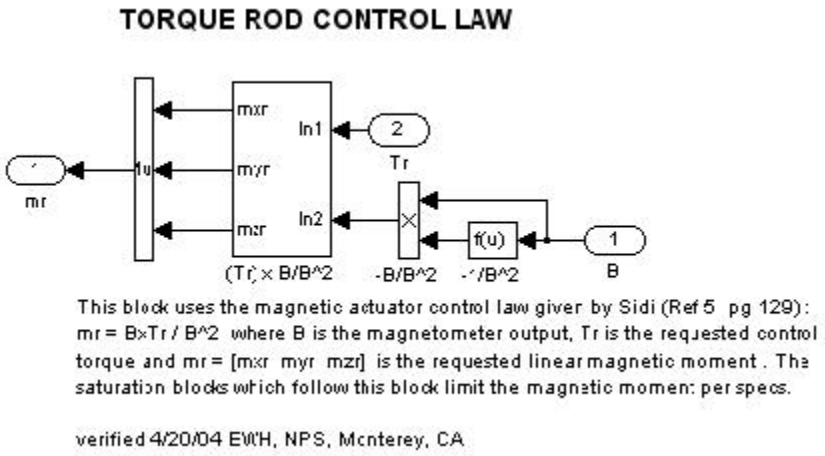
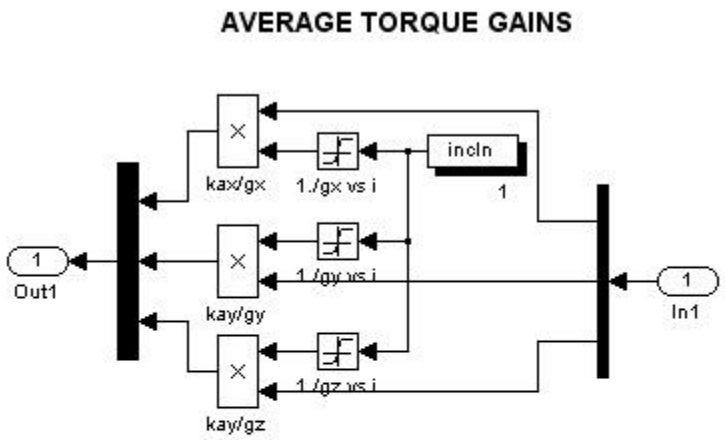


Figure E.9 NPSAT1 Magnetic Moment Sub–Model [After Ref. 9.]

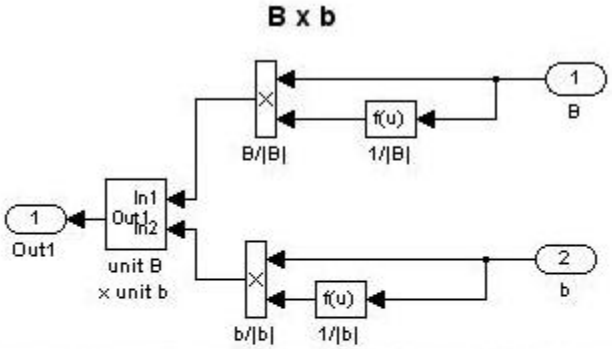


This block attempts to normalize the gain of each channel to unity by dividing by the orbit average gain (as suggested in Ref. below).
The gain ka = [kax kay kaz] in the previous block is a function of moments of inertia. The best value of ka is determined by simulation.

Reference: Martel Et. Al. "Active Magnetic Control System for Gravity Gradient Stabilized Spacecraft", 26-28 September 1988, Logan,UT, pg 7.

Verified 4/19/04 EWH NPS, Monterey, CA

Figure E.10 NPSAT1 Torque Gain Normalization Sub–Model [After Ref. 9.]



This block represents an approximation to the position part of the quaternion control law.
Sidi (Ref 5, pg 156) suggests using e = [q1q4 q2q4 q3q4] as an attitude error vector. The cross Product between unit B and unit b is an approximation to this error vector for magnetic magnetic control systems where: B = magnetometer output and b = predicted magnetic field vector based on emphemeris data and a spherical harmonic model

verified 4/20/04 EWH, NPS, Monterey, CA

Figure E.11 NPSAT1 Attitude Error Sub–Model [After Ref. 9.]

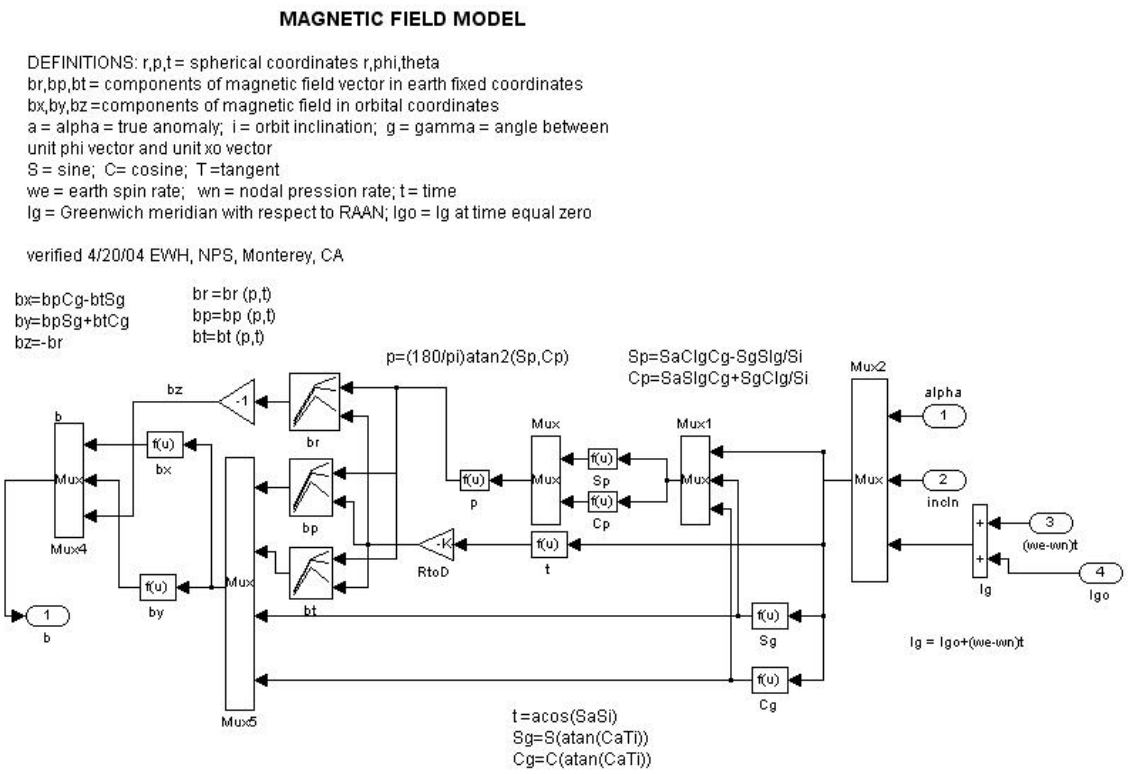


Figure E.12 NPSAT1 External Magnetic Field Ephemeris Table Sub–Model [After Ref. 9.]

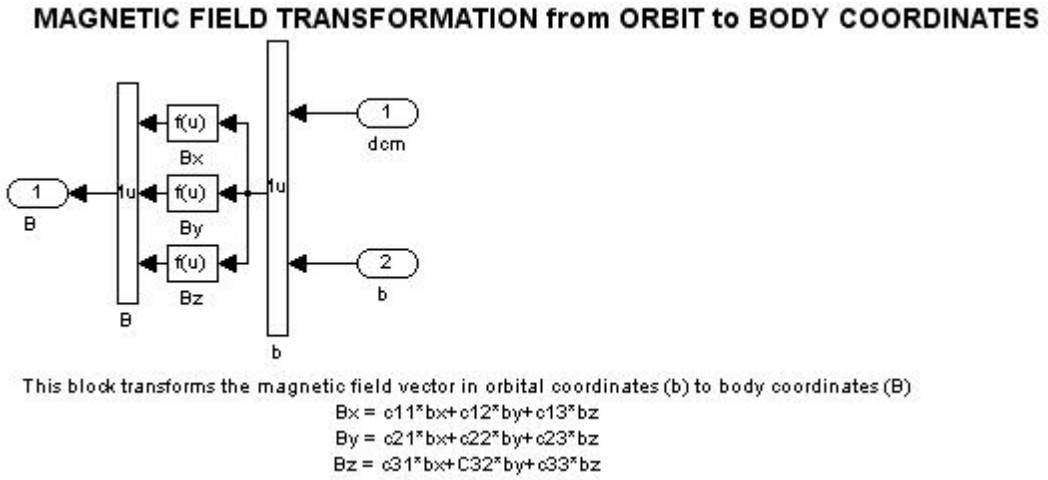
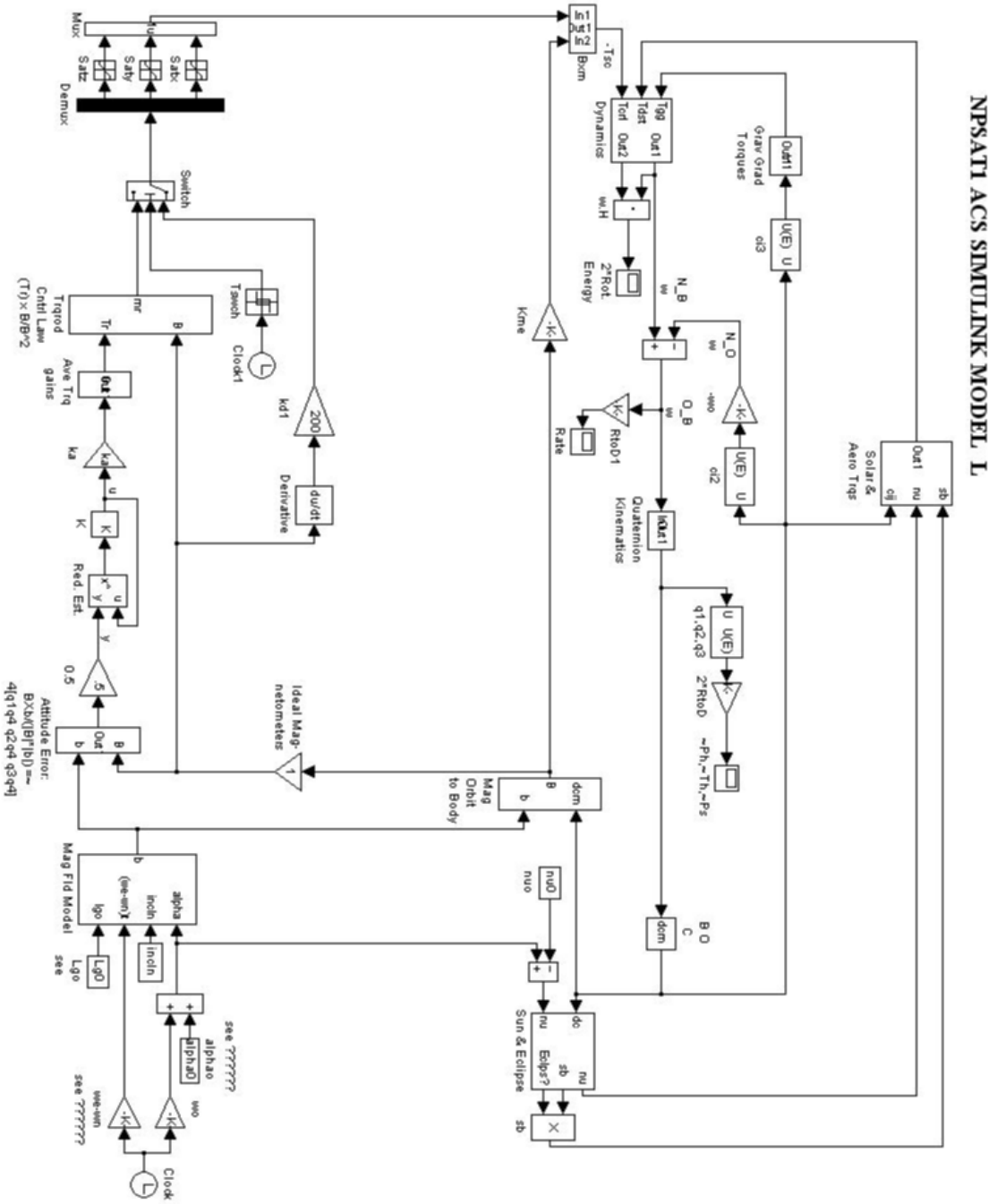


Figure E.13 NPSAT1 Magnetic Field Coordinate Transformation Sub–Model [After Ref. 9.]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. NPSAT1 ACS SIMULINK MODEL L [FROM REF. 9.]

This Appendix displays the SIMULINK model for the NPSAT1 attitude control system that was developed by Leonard and verified by this thesis.



THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. NPSAT1 AIR–BEARING LABORATORY MAGNETIC FIELD SURVEY NO. 1 MATLAB PROGRAM

This Appendix contains the MATLAB program written to determine the coarse magnetic field that resides in the SSAG laboratory.

```
% LCDR Eric W. Herbert, USN, 4/1/04
% Air Bearing Magnetic Field Determination
% Four sets of meausrement values are used. A 5X7 grid corresponds
% to the x and y axis. Z is characterized by 4 heights.
% This algorithm's purpose is to load the measured data,
% remove the magnetic north measurement (not required), convert
% the Honeywell HM2300 Magnetometer cooridinate system to the
% cooridinate system laid out in the lab. The magnetometer readings are
% measured in counts. These are converted to Tesla by the equation:
% (counts(1 gauss/15000 counts)(1 tesla/10000 gauss). Additionally, it will
% identify the number of samples as 180.  This is from approx 10 seconds worth of
% measurements. The samples will be averaged and used to calculate the
% magnetic field at the specific grid point. field= i1j1z1(1:180,1:3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
x=1:8; y=1:6; yy=rot90(y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %FLOOR LEVEL
% load measured values
load i1j1z1.txt; load i2j1z1.txt; load i3j1z1.txt; load i4j1z1.txt; load i5j1z1.txt; load i6j1z1.txt;
load i7j1z1.txt; load i8j1z1.txt
load i1j2z1.txt; load i2j2z1.txt; load i3j2z1.txt; load i4j2z1.txt; load i5j2z1.txt; load i6j2z1.txt;
load i7j2z1.txt; load i8j2z1.txt
load i1j3z1.txt; load i2j3z1.txt; load i3j3z1.txt; load i4j3z1.txt; load i5j3z1.txt; load i6j3z1.txt;
load i7j3z1.txt; load i8j3z1.txt
load i1j4z1.txt; load i2j4z1.txt; load i3j4z1.txt; load i4j4z1.txt; load i5j4z1.txt; load i6j4z1.txt;
load i7j4z1.txt; load i8j4z1.txt
load i1j5z1.txt; load i2j5z1.txt; load i3j5z1.txt; load i4j5z1.txt; load i5j5z1.txt; load i6j5z1.txt;
load i7j5z1.txt; load i8j5z1.txt
load i1j6z1.txt; load i2j6z1.txt; load i3j6z1.txt; load i4j6z1.txt; load i5j6z1.txt; load i6j6z1.txt;
load i7j6z1.txt; load i8j6z1.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% B field calcula-
tions for i1j1z1 through i8j1z1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
n=[mean(i1j1z1(1:180,1:3)) mean(i2j1z1(1:180,1:3)) mean(i3j1z1(1:180,1:3))
mean(i4j1z1(1:180,1:3)) mean(i5j1z1(1:180,1:3)) mean(i6j1z1(1:180,1:3))
mean(i7j1z1(1:180,1:3)) mean(i8j1z1(1:180,1:3));
  mean(i1j2z1(1:180,1:3)) mean(i2j2z1(1:180,1:3)) mean(i3j2z1(1:180,1:3))
mean(i4j2z1(1:180,1:3)) mean(i5j2z1(1:180,1:3)) mean(i6j2z1(1:180,1:3))
mean(i7j2z1(1:180,1:3)) mean(i8j2z1(1:180,1:3));
  mean(i1j3z1(1:180,1:3)) mean(i2j3z1(1:180,1:3)) mean(i3j3z1(1:180,1:3))
mean(i4j3z1(1:180,1:3)) mean(i5j3z1(1:180,1:3)) mean(i6j3z1(1:180,1:3))
mean(i8j3z1(1:180,1:3)) mean(i8j3z1(1:180,1:3));
  mean(i1j4z1(1:180,1:3)) mean(i2j4z1(1:180,1:3)) mean(i3j4z1(1:180,1:3))
mean(i4j4z1(1:180,1:3)) mean(i5j4z1(1:180,1:3)) mean(i6j4z1(1:180,1:3))
mean(i8j4z1(1:180,1:3)) mean(i8j4z1(1:180,1:3));
  mean(i1j5z1(1:180,1:3)) mean(i2j5z1(1:180,1:3)) mean(i3j5z1(1:180,1:3))
mean(i4j5z1(1:180,1:3)) mean(i5j5z1(1:180,1:3)) mean(i6j5z1(1:180,1:3))
mean(i8j5z1(1:180,1:3)) mean(i8j5z1(1:180,1:3));
  mean(i1j6z1(1:180,1:3)) mean(i2j6z1(1:180,1:3)) mean(i3j6z1(1:180,1:3))
mean(i4j6z1(1:180,1:3)) mean(i5j6z1(1:180,1:3)) mean(i6j6z1(1:180,1:3))
mean(i8j6z1(1:180,1:3)) mean(i8j6z1(1:180,1:3))];

B_111=sqrt(n(1,1)^2+n(1,2)^2+n(1,3))/(15000*10000);
B_211=sqrt(n(1,4)^2+n(1,5)^2+n(1,6))/(15000*10000);
B_311=sqrt(n(1,7)^2+n(1,8)^2+n(1,9))/(15000*10000);
B_411=sqrt(n(1,10)^2+n(1,11)^2+n(1,12))/(15000*10000);
B_511=sqrt(n(1,13)^2+n(1,14)^2+n(1,15))/(15000*10000);
B_611=sqrt(n(1,16)^2+n(1,17)^2+n(1,18))/(15000*10000);
B_711=sqrt(n(1,19)^2+n(1,20)^2+n(1,21))/(15000*10000);
B_811=sqrt(n(1,22)^2+n(1,23)^2+n(1,24))/(15000*10000);

B_121=sqrt(n(2,1)^2+n(2,2)^2+n(2,3))/(15000*10000);
B_221=sqrt(n(2,4)^2+n(2,5)^2+n(2,6))/(15000*10000);
B_321=sqrt(n(2,7)^2+n(2,8)^2+n(2,9))/(15000*10000);
B_421=sqrt(n(2,10)^2+n(2,11)^2+n(2,12))/(15000*10000);
B_521=sqrt(n(2,13)^2+n(2,14)^2+n(2,15))/(15000*10000);
B_621=sqrt(n(2,16)^2+n(2,17)^2+n(2,18))/(15000*10000);
B_721=sqrt(n(2,19)^2+n(2,20)^2+n(2,21))/(15000*10000);
B_821=sqrt(n(2,22)^2+n(2,23)^2+n(2,24))/(15000*10000);

B_131=sqrt(n(3,1)^2+n(3,2)^2+n(3,3))/(15000*10000);
B_231=sqrt(n(3,4)^2+n(3,5)^2+n(3,6))/(15000*10000);
B_331=sqrt(n(3,7)^2+n(3,8)^2+n(3,9))/(15000*10000);
B_431=sqrt(n(3,10)^2+n(3,11)^2+n(3,12))/(15000*10000);
B_531=sqrt(n(3,13)^2+n(3,14)^2+n(3,15))/(15000*10000);
B_631=sqrt(n(3,16)^2+n(3,17)^2+n(3,18))/(15000*10000);
B_731=sqrt(n(3,19)^2+n(3,20)^2+n(3,21))/(15000*10000);
B_831=sqrt(n(3,22)^2+n(3,23)^2+n(3,24))/(15000*10000);
```

```
B_141=sqrt(n(4,1)^2+n(4,2)^2+n(4,3))/(15000*10000);
B_241=sqrt(n(4,4)^2+n(4,5)^2+n(4,6))/(15000*10000);
B_341=sqrt(n(4,7)^2+n(4,8)^2+n(4,9))/(15000*10000);
B_441=sqrt(n(4,10)^2+n(4,11)^2+n(4,12))/(15000*10000);
B_541=sqrt(n(4,13)^2+n(4,14)^2+n(4,15))/(15000*10000);
B_641=sqrt(n(4,16)^2+n(4,17)^2+n(4,18))/(15000*10000);
B_741=sqrt(n(4,19)^2+n(4,20)^2+n(4,21))/(15000*10000);
B_841=sqrt(n(4,22)^2+n(4,23)^2+n(4,24))/(15000*10000);

B_151=sqrt(n(5,1)^2+n(5,2)^2+n(5,3))/(15000*10000);
B_251=sqrt(n(5,4)^2+n(5,5)^2+n(5,6))/(15000*10000);
B_351=sqrt(n(5,7)^2+n(5,8)^2+n(5,9))/(15000*10000);
B_451=sqrt(n(5,10)^2+n(5,11)^2+n(5,12))/(15000*10000);
B_551=sqrt(n(5,13)^2+n(5,14)^2+n(5,15))/(15000*10000);
B_651=sqrt(n(5,16)^2+n(5,17)^2+n(5,18))/(15000*10000);
B_751=sqrt(n(5,19)^2+n(5,20)^2+n(5,21))/(15000*10000);
B_851=sqrt(n(5,22)^2+n(5,23)^2+n(5,24))/(15000*10000);

B_161=sqrt(n(6,1)^2+n(6,2)^2+n(6,3))/(15000*10000);
B_261=sqrt(n(6,4)^2+n(6,5)^2+n(6,6))/(15000*10000);
B_361=sqrt(n(6,7)^2+n(6,8)^2+n(6,9))/(15000*10000);
B_461=sqrt(n(6,10)^2+n(6,11)^2+n(6,12))/(15000*10000);
B_561=sqrt(n(6,13)^2+n(6,14)^2+n(6,15))/(15000*10000);
B_661=sqrt(n(6,16)^2+n(6,17)^2+n(6,18))/(15000*10000);
B_761=sqrt(n(6,19)^2+n(6,20)^2+n(6,21))/(15000*10000);
B_861=sqrt(n(6,22)^2+n(6,23)^2+n(6,24))/(15000*10000);


B_level_1=[B_161 B_261 B_361 B_461 B_561 B_661 B_761 B_861;
 % values in Tesla
   B_151 B_251 B_351 B_451 B_551 B_651 B_751 B_851;
   B_141 B_241 B_341 B_441 B_541 B_641 B_741 B_841;
   B_131 B_231 B_331 B_431 B_531 B_631 B_731 B_831;
   B_121 B_221 B_321 B_421 B_521 B_621 B_721 B_821;
   B_111 B_211 B_311 B_411 B_511 B_611 B_711 B_811];
figure(1)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_1);
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
figure(2)
C=contour3(X,Y,B_level_1,10);
clabel(C)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
```

```
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
figure(3)
[DB]=gradient(B_level_1,.5,.5);
D=contour3(X,Y,B_level_1);
clabel(D)
hold on
quiver(B_level_1,DB)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load i1j1z2.txt; load i2j1z2.txt; load i3j1z2.txt; load i4j1z2.txt; load i5j1z2.txt; load i6j1z2.txt;
load i7j1z2.txt; load i8j1z2.txt
load i1j2z2.txt; load i2j2z2.txt; load i3j2z2.txt; load i4j2z2.txt; load i5j2z2.txt; load i6j2z2.txt;
load i7j2z2.txt; load i8j2z2.txt
load i1j3z2.txt; load i2j3z2.txt; load i3j3z2.txt; load i4j3z2.txt; load i5j3z2.txt; load i6j3z2.txt;
load i7j3z2.txt; load i8j3z2.txt
load i1j4z2.txt; load i2j4z2.txt; load i3j4z2.txt; load i4j4z2.txt; load i5j4z2.txt; load i6j4z2.txt;
load i7j4z2.txt; load i8j4z2.txt
load i1j5z2.txt; load i2j5z2.txt; load i3j5z2.txt; load i4j5z2.txt; load i5j5z2.txt; load i6j5z2.txt;
load i7j5z2.txt; load i8j5z2.txt
load i1j6z2.txt; load i2j6z2.txt; load i3j6z2.txt; load i4j6z2.txt; load i5j6z2.txt; load i6j6z2.txt;
load i7j6z2.txt; load i8j6z2.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% B field calculations for i1j1z2 through i8j1z2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%n1=[mean(i1j1z2(1:
180,1:3)) mean(i2j1z2(1:180,1:3)) mean(i3j1z2(1:180,1:3)) mean(i4j1z2(1:180,1:3))
mean(i5j1z2(1:180,1:3)) mean(i6j1z2(1:180,1:3)) mean(i7j1z2(1:180,1:3))
mean(i8j1z2(1:180,1:3));
  mean(i1j2z2(1:180,1:3)) mean(i2j2z2(1:180,1:3)) mean(i3j2z2(1:180,1:3))
mean(i4j2z2(1:180,1:3)) mean(i5j2z2(1:180,1:3)) mean(i6j2z2(1:180,1:3))
mean(i7j2z2(1:180,1:3)) mean(i8j2z2(1:180,1:3));
  mean(i1j3z2(1:180,1:3)) mean(i2j3z2(1:180,1:3)) mean(i3j3z2(1:180,1:3))
mean(i4j3z2(1:180,1:3)) mean(i5j3z2(1:180,1:3)) mean(i6j3z2(1:180,1:3))
mean(i8j3z2(1:180,1:3)) mean(i8j3z2(1:180,1:3));
  mean(i1j4z2(1:180,1:3)) mean(i2j4z2(1:180,1:3)) mean(i3j4z2(1:180,1:3))
mean(i4j4z2(1:180,1:3)) mean(i5j4z2(1:180,1:3)) mean(i6j4z2(1:180,1:3))
mean(i8j4z2(1:180,1:3)) mean(i8j4z2(1:180,1:3));
  mean(i1j5z2(1:180,1:3)) mean(i2j5z2(1:180,1:3)) mean(i3j5z2(1:180,1:3))
mean(i4j5z2(1:180,1:3)) mean(i5j5z2(1:180,1:3)) mean(i6j5z2(1:180,1:3))
mean(i8j5z2(1:180,1:3)) mean(i8j5z2(1:180,1:3));
  mean(i1j6z2(1:180,1:3)) mean(i2j6z2(1:180,1:3)) mean(i3j6z2(1:180,1:3))
mean(i4j6z2(1:180,1:3)) mean(i5j6z2(1:180,1:3)) mean(i6j6z2(1:180,1:3))
mean(i8j6z2(1:180,1:3)) mean(i8j6z2(1:180,1:3))];
```

```
B_112=sqrt(n1(1,1)^2+n1(1,2)^2+n1(1,3))/(15000*10000);
B_212=sqrt(n1(1,4)^2+n1(1,5)^2+n1(1,6))/(15000*10000);
B_312=sqrt(n1(1,7)^2+n1(1,8)^2+n1(1,9))/(15000*10000);
B_412=sqrt(n1(1,10)^2+n1(1,11)^2+n1(1,12))/(15000*10000);
B_512=sqrt(n1(1,13)^2+n1(1,14)^2+n1(1,15))/(15000*10000);
B_612=sqrt(n1(1,16)^2+n1(1,17)^2+n1(1,18))/(15000*10000);
B_712=sqrt(n1(1,19)^2+n1(1,20)^2+n1(1,21))/(15000*10000);
B_812=sqrt(n1(1,22)^2+n1(1,23)^2+n1(1,24))/(15000*10000);

B_122=sqrt(n1(2,1)^2+n1(2,2)^2+n1(2,3))/(15000*10000);
B_222=sqrt(n1(2,4)^2+n1(2,5)^2+n1(2,6))/(15000*10000);
B_322=sqrt(n1(2,7)^2+n1(2,8)^2+n1(2,9))/(15000*10000);
B_422=sqrt(n1(2,10)^2+n1(2,11)^2+n1(2,12))/(15000*10000);
B_522=sqrt(n1(2,13)^2+n1(2,14)^2+n1(2,15))/(15000*10000);
B_622=sqrt(n1(2,16)^2+n1(2,17)^2+n1(2,18))/(15000*10000);
B_722=sqrt(n1(2,19)^2+n1(2,20)^2+n1(2,21))/(15000*10000);
B_822=sqrt(n1(2,22)^2+n1(2,23)^2+n1(2,24))/(15000*10000);

B_132=sqrt(n1(3,1)^2+n1(3,2)^2+n1(3,3))/(15000*10000);
B_232=sqrt(n1(3,4)^2+n1(3,5)^2+n1(3,6))/(15000*10000);
B_332=sqrt(n1(3,7)^2+n1(3,8)^2+n1(3,9))/(15000*10000);
B_432=sqrt(n1(3,10)^2+n1(3,11)^2+n1(3,12))/(15000*10000);
B_532=sqrt(n1(3,13)^2+n1(3,14)^2+n1(3,15))/(15000*10000);
B_632=sqrt(n1(3,16)^2+n1(3,17)^2+n1(3,18))/(15000*10000);
B_732=sqrt(n1(3,19)^2+n1(3,20)^2+n1(3,21))/(15000*10000);
B_832=sqrt(n1(3,22)^2+n1(3,23)^2+n1(3,24))/(15000*10000);

B_142=sqrt(n1(4,1)^2+n1(4,2)^2+n1(4,3))/(15000*10000);
B_242=sqrt(n1(4,4)^2+n1(4,5)^2+n1(4,6))/(15000*10000);
B_342=sqrt(n1(4,7)^2+n1(4,8)^2+n1(4,9))/(15000*10000);
B_442=sqrt(n1(4,10)^2+n1(4,11)^2+n1(4,12))/(15000*10000);
B_542=sqrt(n1(4,13)^2+n1(4,14)^2+n1(4,15))/(15000*10000);
B_642=sqrt(n1(4,16)^2+n1(4,17)^2+n1(4,18))/(15000*10000);
B_742=sqrt(n1(4,19)^2+n1(4,20)^2+n1(4,21))/(15000*10000);
B_842=sqrt(n1(4,22)^2+n1(4,23)^2+n1(4,24))/(15000*10000);

B_152=sqrt(n1(5,1)^2+n1(5,2)^2+n1(5,3))/(15000*10000);
B_252=sqrt(n1(5,4)^2+n1(5,5)^2+n1(5,6))/(15000*10000);
B_352=sqrt(n1(5,7)^2+n1(5,8)^2+n1(5,9))/(15000*10000);
B_452=sqrt(n1(5,10)^2+n1(5,11)^2+n1(5,12))/(15000*10000);
B_552=sqrt(n1(5,13)^2+n1(5,14)^2+n1(5,15))/(15000*10000);
B_652=sqrt(n1(5,16)^2+n1(5,17)^2+n1(5,18))/(15000*10000);
B_752=sqrt(n1(5,19)^2+n1(5,20)^2+n1(5,21))/(15000*10000);
B_852=sqrt(n1(5,22)^2+n1(5,23)^2+n1(5,24))/(15000*10000);
```

```
B_162=sqrt(n1(6,1)^2+n1(6,2)^2+n1(6,3))/(15000*10000);
B_262=sqrt(n1(6,4)^2+n1(6,5)^2+n1(6,6))/(15000*10000);
B_362=sqrt(n1(6,7)^2+n1(6,8)^2+n1(6,9))/(15000*10000);
B_462=sqrt(n1(6,10)^2+n1(6,11)^2+n1(6,12))/(15000*10000);
B_562=sqrt(n1(6,13)^2+n1(6,14)^2+n1(6,15))/(15000*10000);
B_662=sqrt(n1(6,16)^2+n1(6,17)^2+n1(6,18))/(15000*10000);
B_762=sqrt(n1(6,19)^2+n1(6,20)^2+n1(6,21))/(15000*10000);
B_862=sqrt(n1(6,22)^2+n1(6,23)^2+n1(6,24))/(15000*10000);

B_level_2=[B_162 B_262 B_362 B_462 B_562 B_662 B_762 B_862;
% values in Tesla
     B_152 B_252 B_352 B_452 B_552 B_652 B_752 B_852;
     B_142 B_242 B_342 B_442 B_542 B_642 B_742 B_842;
     B_132 B_232 B_332 B_432 B_532 B_632 B_732 B_832;
     B_122 B_222 B_322 B_422 B_522 B_622 B_722 B_822;
     B_112 B_212 B_312 B_412 B_512 B_612 B_712 B_812];
figure(4)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_2)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')
figure(5)
C2=contour3(X,Y,B_level_2,10);
clabel(C2)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')
figure(6)
[DB2]=gradient(B_level_2,.5,.5);
D2=contour3(X,Y,B_level_2);
clabel(D2)
hold on
quiver(B_level_2,DB2)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%B field calculations for i1j1z3 through i8j1z3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load i1j1z3.txt; load i2j1z3.txt; load i3j1z3.txt; load i4j1z3.txt; load i5j1z3.txt; load i6j1z3.txt;
load i7j1z3.txt; load i8j1z3.txt
```

```
load i1j2z3.txt; load i2j2z3.txt; load i3j2z3.txt; load i4j2z3.txt; load i5j2z3.txt; load i6j2z3.txt;
load i7j2z3.txt; load i8j2z3.txt
load i1j3z3.txt; load i2j3z3.txt; load i3j3z3.txt; load i4j3z3.txt; load i5j3z3.txt; load i6j3z3.txt;
load i7j3z3.txt; load i8j3z3.txt
load i1j4z3.txt; load i2j4z3.txt; load i3j4z3.txt; load i4j4z3.txt; load i5j4z3.txt; load i6j4z3.txt;
load i7j4z3.txt; load i8j4z3.txt
load i1j5z3.txt; load i2j5z3.txt; load i3j5z3.txt; load i4j5z3.txt; load i5j5z3.txt; load i6j5z3.txt;
load i7j5z3.txt; load i8j5z3.txt
load i1j6z3.txt; load i2j6z3.txt; load i3j6z3.txt; load i4j6z3.txt; load i5j6z3.txt; load i6j6z3.txt;
load i7j6z3.txt; load i8j6z3.txt

n3=[mean(i1j1z3(1:180,1:3)) mean(i2j1z3(1:180,1:3)) mean(i3j1z3(1:180,1:3))
mean(i4j1z3(1:180,1:3)) mean(i5j1z3(1:180,1:3)) mean(i6j1z3(1:180,1:3))
mean(i7j1z3(1:180,1:3)) mean(i8j1z3(1:180,1:3));
  mean(i1j2z3(1:180,1:3)) mean(i2j2z3(1:180,1:3)) mean(i3j2z3(1:180,1:3))
mean(i4j2z3(1:180,1:3)) mean(i5j2z3(1:180,1:3)) mean(i6j2z3(1:180,1:3))
mean(i7j2z3(1:180,1:3)) mean(i8j2z3(1:160,1:3));
  mean(i1j3z3(1:180,1:3)) mean(i2j3z3(1:180,1:3)) mean(i3j3z3(1:180,1:3))
mean(i4j3z3(1:180,1:3)) mean(i5j3z3(1:170,1:3)) mean(i6j3z3(1:180,1:3))
mean(i8j3z3(1:180,1:3)) mean(i8j3z3(1:180,1:3));
  mean(i1j4z3(1:180,1:3)) mean(i2j4z3(1:180,1:3)) mean(i3j4z3(1:180,1:3))
mean(i4j4z3(1:180,1:3)) mean(i5j4z3(1:180,1:3)) mean(i6j4z3(1:180,1:3))
mean(i8j4z3(1:180,1:3)) mean(i8j4z3(1:180,1:3));
  mean(i1j5z3(1:180,1:3)) mean(i2j5z3(1:180,1:3)) mean(i3j5z3(1:180,1:3))
mean(i4j5z3(1:180,1:3)) mean(i5j5z3(1:180,1:3)) mean(i6j5z3(1:180,1:3))
mean(i8j5z3(1:180,1:3)) mean(i8j5z3(1:180,1:3));
  mean(i1j6z3(1:180,1:3)) mean(i2j6z3(1:180,1:3)) mean(i3j6z3(1:180,1:3))
mean(i4j6z3(1:180,1:3)) mean(i5j6z3(1:180,1:3)) mean(i6j6z3(1:180,1:3))
mean(i8j6z3(1:180,1:3)) mean(i8j6z3(1:180,1:3))];

B_113=sqrt(n3(1,1)^2+n3(1,2)^2+n3(1,3))/(15000*10000);
B_213=sqrt(n3(1,4)^2+n3(1,5)^2+n3(1,6))/(15000*10000);
B_313=sqrt(n3(1,7)^2+n3(1,8)^2+n3(1,9))/(15000*10000);
B_413=sqrt(n3(1,10)^2+n3(1,11)^2+n3(1,12))/(15000*10000);
B_513=sqrt(n3(1,13)^2+n3(1,14)^2+n3(1,15))/(15000*10000);
B_613=sqrt(n3(1,16)^2+n3(1,17)^2+n3(1,18))/(15000*10000);
B_713=sqrt(n3(1,19)^2+n3(1,20)^2+n3(1,21))/(15000*10000);
B_813=sqrt(n3(1,22)^2+n3(1,23)^2+n3(1,24))/(15000*10000);

B_123=sqrt(n3(2,1)^2+n3(2,2)^2+n3(2,3))/(15000*10000);
B_223=sqrt(n3(2,4)^2+n3(2,5)^2+n3(2,6))/(15000*10000);
B_323=sqrt(n3(2,7)^2+n3(2,8)^2+n3(2,9))/(15000*10000);
B_423=sqrt(n3(2,10)^2+n3(2,11)^2+n3(2,12))/(15000*10000);
B_523=sqrt(n3(2,13)^2+n3(2,14)^2+n3(2,15))/(15000*10000);
B_623=sqrt(n3(2,16)^2+n3(2,17)^2+n3(2,18))/(15000*10000);
```

```
B_723=sqrt(n3(2,19)^2+n3(2,20)^2+n3(2,21))/(15000*10000);
B_823=sqrt(n3(2,22)^2+n3(2,23)^2+n3(2,24))/(15000*10000);

B_133=sqrt(n3(3,1)^2+n3(3,2)^2+n3(3,3))/(15000*10000);
B_233=sqrt(n3(3,4)^2+n3(3,5)^2+n3(3,6))/(15000*10000);
B_333=sqrt(n3(3,7)^2+n3(3,8)^2+n3(3,9))/(15000*10000);
B_433=sqrt(n3(3,10)^2+n3(3,11)^2+n3(3,12))/(15000*10000);
B_533=sqrt(n3(3,13)^2+n3(3,14)^2+n3(3,15))/(15000*10000);
B_633=sqrt(n3(3,16)^2+n3(3,17)^2+n3(3,18))/(15000*10000);
B_733=sqrt(n3(3,19)^2+n3(3,20)^2+n3(3,21))/(15000*10000);
B_833=sqrt(n3(3,22)^2+n3(3,23)^2+n3(3,24))/(15000*10000);

B_143=sqrt(n3(4,1)^2+n3(4,2)^2+n3(4,3))/(15000*10000);
B_243=sqrt(n3(4,4)^2+n3(4,5)^2+n3(4,6))/(15000*10000);
B_343=sqrt(n3(4,7)^2+n3(4,8)^2+n3(4,9))/(15000*10000);
B_443=sqrt(n3(4,10)^2+n3(4,11)^2+n3(4,12))/(15000*10000);
B_543=sqrt(n3(4,13)^2+n3(4,14)^2+n3(4,15))/(15000*10000);
B_643=sqrt(n3(4,16)^2+n3(4,17)^2+n3(4,18))/(15000*10000);
B_743=sqrt(n3(4,19)^2+n3(4,20)^2+n3(4,21))/(15000*10000);
B_843=sqrt(n3(4,22)^2+n3(4,23)^2+n3(4,24))/(15000*10000);

B_153=sqrt(n3(5,1)^2+n3(5,2)^2+n3(5,3))/(15000*10000);
B_253=sqrt(n3(5,4)^2+n3(5,5)^2+n3(5,6))/(15000*10000);
B_353=sqrt(n3(5,7)^2+n3(5,8)^2+n3(5,9))/(15000*10000);
B_453=sqrt(n3(5,10)^2+n3(5,11)^2+n3(5,12))/(15000*10000);
B_553=sqrt(n3(5,13)^2+n3(5,14)^2+n3(5,15))/(15000*10000);
B_653=sqrt(n3(5,16)^2+n3(5,17)^2+n3(5,18))/(15000*10000);
B_753=sqrt(n3(5,19)^2+n3(5,20)^2+n3(5,21))/(15000*10000);
B_853=sqrt(n3(5,22)^2+n3(5,23)^2+n3(5,24))/(15000*10000);

B_163=sqrt(n3(6,1)^2+n3(6,2)^2+n3(6,3))/(15000*10000);
B_263=sqrt(n3(6,4)^2+n3(6,5)^2+n3(6,6))/(15000*10000);
B_363=sqrt(n3(6,7)^2+n3(6,8)^2+n3(6,9))/(15000*10000);
B_463=sqrt(n3(6,10)^2+n3(6,11)^2+n3(6,12))/(15000*10000);
B_563=sqrt(n3(6,13)^2+n3(6,14)^2+n3(6,15))/(15000*10000);
B_663=sqrt(n3(6,16)^2+n3(6,17)^2+n3(6,18))/(15000*10000);
B_763=sqrt(n3(6,19)^2+n3(6,20)^2+n3(6,21))/(15000*10000);
B_863=sqrt(n3(6,22)^2+n3(6,23)^2+n3(6,24))/(15000*10000);

B_level_3=[B_163 B_263 B_363 B_463 B_563 B_663 B_763 B_863;
% values in Tesla
        B_153 B_253 B_353 B_453 B_553 B_653 B_753 B_853;
        B_143 B_243 B_343 B_443 B_543 B_643 B_743 B_843;
        B_133 B_233 B_333 B_433 B_533 B_633 B_733 B_833;
        B_123 B_223 B_323 B_423 B_523 B_623 B_723 B_823;
```

```
        B_113 B_213 B_313 B_413 B_513 B_613 B_713 B_813];
figure(7)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_3)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')
figure(8)
C3=contour3(X,Y,B_level_3,10);
clabel(C3)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')
figure(9)
[DB3]=gradient(B_level_3,.5,.5);
D3=contour3(X,Y,B_level_3);
clabel(D3)
hold on
quiver(B_level_3,DB3)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')

%B field calculations for i1j1z4 through i8j1z4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load i1j1z4.txt; load i2j1z4.txt; load i3j1z4.txt; load i4j1z4.txt; load i5j1z4.txt; load i6j1z4.txt;
load i7j1z4.txt; load i8j1z4.txt
load i1j2z4.txt; load i2j2z4.txt; load i3j2z4.txt; load i4j2z4.txt; load i5j2z4.txt; load i6j2z4.txt;
load i7j2z4.txt; load i8j2z4.txt
load i1j3z4.txt; load i2j3z4.txt; load i3j3z4.txt; load i4j3z4.txt; load i5j3z4.txt; load i6j3z4.txt;
load i7j3z4.txt; load i8j3z4.txt
load i1j4z4.txt; load i2j4z4.txt; load i3j4z4.txt; load i4j4z4.txt; load i5j4z4.txt; load i6j4z4.txt;
load i7j4z4.txt; load i8j4z4.txt
load i1j5z4.txt; load i2j5z4.txt; load i3j5z4.txt; load i4j5z4.txt; load i5j5z4.txt; load i6j5z4.txt;
load i7j5z4.txt; load i8j5z4.txt
load i1j6z4.txt; load i2j6z4.txt; load i3j6z4.txt; load i4j6z4.txt; load i5j6z4.txt; load i6j6z4.txt;
load i7j6z4.txt; load i8j6z4.txt

n4=[mean(i1j1z4(1:180,1:3)) mean(i2j1z4(1:180,1:3)) mean(i3j1z4(1:180,1:3))
mean(i4j1z4(1:180,1:3)) mean(i5j1z4(1:180,1:3)) mean(i6j1z4(1:180,1:3))
mean(i7j1z4(1:180,1:3)) mean(i8j1z4(1:180,1:3));
  mean(i1j2z4(1:180,1:3)) mean(i2j2z4(1:170,1:3)) mean(i3j2z4(1:180,1:3))
mean(i4j2z4(1:180,1:3)) mean(i5j2z4(1:180,1:3)) mean(i6j2z4(1:180,1:3))
mean(i7j2z4(1:180,1:3)) mean(i8j2z4(1:160,1:3));
```

```
   mean(i1j3z4(1:180,1:3)) mean(i2j3z4(1:180,1:3)) mean(i3j3z4(1:180,1:3))
mean(i4j3z4(1:180,1:3)) mean(i5j3z4(1:170,1:3)) mean(i6j3z4(1:180,1:3))
mean(i8j3z4(1:180,1:3)) mean(i8j3z4(1:180,1:3));
   mean(i1j4z4(1:180,1:3)) mean(i2j4z4(1:180,1:3)) mean(i3j4z4(1:180,1:3))
mean(i4j4z4(1:180,1:3)) mean(i5j4z4(1:180,1:3)) mean(i6j4z4(1:180,1:3))
mean(i8j4z4(1:180,1:3)) mean(i8j4z4(1:180,1:3));
   mean(i1j5z4(1:180,1:3)) mean(i2j5z4(1:180,1:3)) mean(i3j5z4(1:180,1:3))
mean(i4j5z4(1:180,1:3)) mean(i5j5z4(1:180,1:3)) mean(i6j5z4(1:180,1:3))
mean(i8j5z4(1:180,1:3)) mean(i8j5z4(1:180,1:3));
   mean(i1j6z4(1:180,1:3)) mean(i2j6z4(1:180,1:3)) mean(i3j6z4(1:180,1:3))
mean(i4j6z4(1:180,1:3)) mean(i5j6z4(1:180,1:3)) mean(i6j6z4(1:180,1:3))
mean(i8j6z4(1:180,1:3)) mean(i8j6z4(1:180,1:3))];

B_114=sqrt(n4(1,1)^2+n4(1,2)^2+n4(1,3))/(15000*10000);
B_214=sqrt(n4(1,4)^2+n4(1,5)^2+n4(1,6))/(15000*10000);
B_314=sqrt(n4(1,7)^2+n4(1,8)^2+n4(1,9))/(15000*10000);
B_414=sqrt(n4(1,10)^2+n4(1,11)^2+n4(1,12))/(15000*10000);
B_514=sqrt(n4(1,13)^2+n4(1,14)^2+n4(1,15))/(15000*10000);
B_614=sqrt(n4(1,16)^2+n4(1,17)^2+n4(1,18))/(15000*10000);
B_714=sqrt(n4(1,19)^2+n4(1,20)^2+n4(1,21))/(15000*10000);
B_814=sqrt(n4(1,22)^2+n4(1,23)^2+n4(1,24))/(15000*10000);

B_124=sqrt(n4(2,1)^2+n4(2,2)^2+n4(2,3))/(15000*10000);
B_224=sqrt(n4(2,4)^2+n4(2,5)^2+n4(2,6))/(15000*10000);
B_324=sqrt(n4(2,7)^2+n4(2,8)^2+n4(2,9))/(15000*10000);
B_424=sqrt(n4(2,10)^2+n4(2,11)^2+n4(2,12))/(15000*10000);
B_524=sqrt(n4(2,13)^2+n4(2,14)^2+n4(2,15))/(15000*10000);
B_624=sqrt(n4(2,16)^2+n4(2,17)^2+n4(2,18))/(15000*10000);
B_724=sqrt(n4(2,19)^2+n4(2,20)^2+n4(2,21))/(15000*10000);
B_824=sqrt(n4(2,22)^2+n4(2,23)^2+n4(2,24))/(15000*10000);

B_134=sqrt(n4(3,1)^2+n4(3,2)^2+n4(3,3))/(15000*10000);
B_234=sqrt(n4(3,4)^2+n4(3,5)^2+n4(3,6))/(15000*10000);
B_334=sqrt(n4(3,7)^2+n4(3,8)^2+n4(3,9))/(15000*10000);
B_434=sqrt(n4(3,10)^2+n4(3,11)^2+n4(3,12))/(15000*10000);
B_534=sqrt(n4(3,13)^2+n4(3,14)^2+n4(3,15))/(15000*10000);
B_634=sqrt(n4(3,16)^2+n4(3,17)^2+n4(3,18))/(15000*10000);
B_734=sqrt(n4(3,19)^2+n4(3,20)^2+n4(3,21))/(15000*10000);
B_834=sqrt(n4(3,22)^2+n4(3,23)^2+n4(3,24))/(15000*10000);

B_144=sqrt(n4(4,1)^2+n4(4,2)^2+n4(4,3))/(15000*10000);
B_244=sqrt(n4(4,4)^2+n4(4,5)^2+n4(4,6))/(15000*10000);
B_344=sqrt(n4(4,7)^2+n4(4,8)^2+n4(4,9))/(15000*10000);
```

```
B_444=sqrt(n4(4,10)^2+n4(4,11)^2+n4(4,12))/(15000*10000);
B_544=sqrt(n4(4,13)^2+n4(4,14)^2+n4(4,15))/(15000*10000);
B_644=sqrt(n4(4,16)^2+n4(4,17)^2+n4(4,18))/(15000*10000);
B_744=sqrt(n4(4,19)^2+n4(4,20)^2+n4(4,21))/(15000*10000);
B_844=sqrt(n4(4,22)^2+n4(4,23)^2+n4(4,24))/(15000*10000);

B_154=sqrt(n4(5,1)^2+n4(5,2)^2+n4(5,3))/(15000*10000);
B_254=sqrt(n4(5,4)^2+n4(5,5)^2+n4(5,6))/(15000*10000);
B_354=sqrt(n4(5,7)^2+n4(5,8)^2+n4(5,9))/(15000*10000);
B_454=sqrt(n4(5,10)^2+n4(5,11)^2+n4(5,12))/(15000*10000);
B_554=sqrt(n4(5,13)^2+n4(5,14)^2+n4(5,15))/(15000*10000);
B_654=sqrt(n4(5,16)^2+n4(5,17)^2+n4(5,18))/(15000*10000);
B_754=sqrt(n4(5,19)^2+n4(5,20)^2+n4(5,21))/(15000*10000);
B_854=sqrt(n4(5,22)^2+n4(5,23)^2+n4(5,24))/(15000*10000);

B_164=sqrt(n4(6,1)^2+n4(6,2)^2+n4(6,3))/(15000*10000);
B_264=sqrt(n4(6,4)^2+n4(6,5)^2+n4(6,6))/(15000*10000);
B_364=sqrt(n4(6,7)^2+n4(6,8)^2+n4(6,9))/(15000*10000);
B_464=sqrt(n4(6,10)^2+n4(6,11)^2+n4(6,12))/(15000*10000);
B_564=sqrt(n4(6,13)^2+n4(6,14)^2+n4(6,15))/(15000*10000);
B_664=sqrt(n4(6,16)^2+n4(6,17)^2+n4(6,18))/(15000*10000);
B_764=sqrt(n4(6,19)^2+n4(6,20)^2+n4(6,21))/(15000*10000);
B_864=sqrt(n4(6,22)^2+n4(6,23)^2+n4(6,24))/(15000*10000);

B_level_4=[B_164 B_264 B_364 B_464 B_564 B_664 B_764 B_864;
% values in Tesla
        B_154 B_254 B_354 B_454 B_554 B_654 B_754 B_854;
        B_144 B_244 B_344 B_444 B_544 B_644 B_744 B_844;
        B_134 B_234 B_334 B_434 B_534 B_634 B_734 B_834;
        B_124 B_224 B_324 B_424 B_524 B_624 B_724 B_824;
        B_114 B_214 B_314 B_414 B_514 B_614 B_714 B_814];
figure(10)
[X,Y]=meshgrid(x,yy)
surf(X,Y,B_level_4)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
figure(11)
C4=contour3(X,Y,B_level_4,10);
clabel(C4)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
figure(12)
[DB4]=gradient(B_level_4,.5,.5);
```

```
D4=contour3(X,Y,B_level_4);
clabel(D4)
hold on
quiver(B_level_4,DB4)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
```

APPENDIX H. NPSAT1 AIR–BEARING LABORATORY MAGNETIC FIELD SURVEY NO. 2 MATLAB PROGRAM

This Appendix contains the second MATLAB program written to determine the coarse magnetic field that resides in the SSAG laboratory.

```
% LCDR Eric W. Herbert, USN, 4/1/04
% Air Bearing Magnetic Field Determination
% Four sets of meausrement values are used. A 5X7 grid corresponds
% to the x and y axis. Z is characterized by 4 heights.
% This algorithm's purpose is to load the measured data,
% remove the magnetic north measurement (not required), convert
% the Honeywell HM2300 Magnetometer cooridinate system to the
% cooridinate system laid out in the lab. The magnetometer readings are
% measured in counts. These are converted to Tesla by the equation:
% (counts(1 gauss/15000 counts)(1 tesla/10000 gauss). Additionally, it will
% identify the number of samples as 180.  This is from approx 10 seconds worth of
% measurements. The samples will be averaged and used to calculate the
% magnetic field at the specific grid point. field= i1j1z1(1:180,1:3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
x=1:8; y=1:6; yy=rot90(y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %FLOOR LEVEL
% load measured values
load i1j1z1.txt; load i2j1z1.txt; load i3j1z1.txt; load i4j1z1.txt; load i5j1z1.txt; load i6j1z1.txt;
load i7j1z1.txt; load i8j1z1.txt
load i1j2z1.txt; load i2j2z1.txt; load i3j2z1.txt; load i4j2z1.txt; load i5j2z1.txt; load i6j2z1.txt;
load i7j2z1.txt; load i8j2z1.txt
load i1j3z1.txt; load i2j3z1.txt; load i3j3z1.txt; load i4j3z1.txt; load i5j3z1.txt; load i6j3z1.txt;
load i7j3z1.txt; load i8j3z1.txt
load i1j4z1.txt; load i2j4z1.txt; load i3j4z1.txt; load i4j4z1.txt; load i5j4z1.txt; load i6j4z1.txt;
load i7j4z1.txt; load i8j4z1.txt
load i1j5z1.txt; load i2j5z1.txt; load i3j5z1.txt; load i4j5z1.txt; load i5j5z1.txt; load i6j5z1.txt;
load i7j5z1.txt; load i8j5z1.txt
load i1j6z1.txt; load i2j6z1.txt; load i3j6z1.txt; load i4j6z1.txt; load i5j6z1.txt; load i6j6z1.txt;
load i7j6z1.txt; load i8j6z1.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% B field calcula-
tions for i1j1z1 through i8j1z1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
n=[mean(i1j1z1(1:180,1:3)) mean(i2j1z1(1:180,1:3)) mean(i3j1z1(1:180,1:3))
mean(i4j1z1(1:180,1:3)) mean(i5j1z1(1:180,1:3)) mean(i6j1z1(1:180,1:3))
mean(i7j1z1(1:180,1:3)) mean(i8j1z1(1:180,1:3));
   mean(i1j2z1(1:180,1:3)) mean(i2j2z1(1:180,1:3)) mean(i3j2z1(1:180,1:3))
mean(i4j2z1(1:180,1:3)) mean(i5j2z1(1:180,1:3)) mean(i6j2z1(1:180,1:3))
mean(i7j2z1(1:180,1:3)) mean(i8j2z1(1:180,1:3));
   mean(i1j3z1(1:180,1:3)) mean(i2j3z1(1:180,1:3)) mean(i3j3z1(1:180,1:3))
mean(i4j3z1(1:180,1:3)) mean(i5j3z1(1:180,1:3)) mean(i6j3z1(1:180,1:3))
mean(i8j3z1(1:180,1:3)) mean(i8j3z1(1:180,1:3));
   mean(i1j4z1(1:180,1:3)) mean(i2j4z1(1:180,1:3)) mean(i3j4z1(1:180,1:3))
mean(i4j4z1(1:180,1:3)) mean(i5j4z1(1:180,1:3)) mean(i6j4z1(1:180,1:3))
mean(i8j4z1(1:180,1:3)) mean(i8j4z1(1:180,1:3));
   mean(i1j5z1(1:180,1:3)) mean(i2j5z1(1:180,1:3)) mean(i3j5z1(1:180,1:3))
mean(i4j5z1(1:180,1:3)) mean(i5j5z1(1:180,1:3)) mean(i6j5z1(1:180,1:3))
mean(i8j5z1(1:180,1:3)) mean(i8j5z1(1:180,1:3));
   mean(i1j6z1(1:180,1:3)) mean(i2j6z1(1:180,1:3)) mean(i3j6z1(1:180,1:3))
mean(i4j6z1(1:180,1:3)) mean(i5j6z1(1:180,1:3)) mean(i6j6z1(1:180,1:3))
mean(i8j6z1(1:180,1:3)) mean(i8j6z1(1:180,1:3))];

B_111=sqrt(n(1,1)^2+n(1,2)^2+n(1,3))/(15000*10000);
B_211=sqrt(n(1,4)^2+n(1,5)^2+n(1,6))/(15000*10000);
B_311=sqrt(n(1,7)^2+n(1,8)^2+n(1,9))/(15000*10000);
B_411=sqrt(n(1,10)^2+n(1,11)^2+n(1,12))/(15000*10000);
B_511=sqrt(n(1,13)^2+n(1,14)^2+n(1,15))/(15000*10000);
B_611=sqrt(n(1,16)^2+n(1,17)^2+n(1,18))/(15000*10000);
B_711=sqrt(n(1,19)^2+n(1,20)^2+n(1,21))/(15000*10000);
B_811=sqrt(n(1,22)^2+n(1,23)^2+n(1,24))/(15000*10000);

B_121=sqrt(n(2,1)^2+n(2,2)^2+n(2,3))/(15000*10000);
B_221=sqrt(n(2,4)^2+n(2,5)^2+n(2,6))/(15000*10000);
B_321=sqrt(n(2,7)^2+n(2,8)^2+n(2,9))/(15000*10000);
B_421=sqrt(n(2,10)^2+n(2,11)^2+n(2,12))/(15000*10000);
B_521=sqrt(n(2,13)^2+n(2,14)^2+n(2,15))/(15000*10000);
B_621=sqrt(n(2,16)^2+n(2,17)^2+n(2,18))/(15000*10000);
B_721=sqrt(n(2,19)^2+n(2,20)^2+n(2,21))/(15000*10000);
B_821=sqrt(n(2,22)^2+n(2,23)^2+n(2,24))/(15000*10000);

B_131=sqrt(n(3,1)^2+n(3,2)^2+n(3,3))/(15000*10000);
B_231=sqrt(n(3,4)^2+n(3,5)^2+n(3,6))/(15000*10000);
B_331=sqrt(n(3,7)^2+n(3,8)^2+n(3,9))/(15000*10000);
B_431=sqrt(n(3,10)^2+n(3,11)^2+n(3,12))/(15000*10000);
B_531=sqrt(n(3,13)^2+n(3,14)^2+n(3,15))/(15000*10000);
B_631=sqrt(n(3,16)^2+n(3,17)^2+n(3,18))/(15000*10000);
B_731=sqrt(n(3,19)^2+n(3,20)^2+n(3,21))/(15000*10000);
B_831=sqrt(n(3,22)^2+n(3,23)^2+n(3,24))/(15000*10000);
```

```
B_141=sqrt(n(4,1)^2+n(4,2)^2+n(4,3))/(15000*10000);
B_241=sqrt(n(4,4)^2+n(4,5)^2+n(4,6))/(15000*10000);
B_341=sqrt(n(4,7)^2+n(4,8)^2+n(4,9))/(15000*10000);
B_441=sqrt(n(4,10)^2+n(4,11)^2+n(4,12))/(15000*10000);
B_541=sqrt(n(4,13)^2+n(4,14)^2+n(4,15))/(15000*10000);
B_641=sqrt(n(4,16)^2+n(4,17)^2+n(4,18))/(15000*10000);
B_741=sqrt(n(4,19)^2+n(4,20)^2+n(4,21))/(15000*10000);
B_841=sqrt(n(4,22)^2+n(4,23)^2+n(4,24))/(15000*10000);

B_151=sqrt(n(5,1)^2+n(5,2)^2+n(5,3))/(15000*10000);
B_251=sqrt(n(5,4)^2+n(5,5)^2+n(5,6))/(15000*10000);
B_351=sqrt(n(5,7)^2+n(5,8)^2+n(5,9))/(15000*10000);
B_451=sqrt(n(5,10)^2+n(5,11)^2+n(5,12))/(15000*10000);
B_551=sqrt(n(5,13)^2+n(5,14)^2+n(5,15))/(15000*10000);
B_651=sqrt(n(5,16)^2+n(5,17)^2+n(5,18))/(15000*10000);
B_751=sqrt(n(5,19)^2+n(5,20)^2+n(5,21))/(15000*10000);
B_851=sqrt(n(5,22)^2+n(5,23)^2+n(5,24))/(15000*10000);

B_161=sqrt(n(6,1)^2+n(6,2)^2+n(6,3))/(15000*10000);
B_261=sqrt(n(6,4)^2+n(6,5)^2+n(6,6))/(15000*10000);
B_361=sqrt(n(6,7)^2+n(6,8)^2+n(6,9))/(15000*10000);
B_461=sqrt(n(6,10)^2+n(6,11)^2+n(6,12))/(15000*10000);
B_561=sqrt(n(6,13)^2+n(6,14)^2+n(6,15))/(15000*10000);
B_661=sqrt(n(6,16)^2+n(6,17)^2+n(6,18))/(15000*10000);
B_761=sqrt(n(6,19)^2+n(6,20)^2+n(6,21))/(15000*10000);
B_861=sqrt(n(6,22)^2+n(6,23)^2+n(6,24))/(15000*10000);


B_level_1=[B_161 B_261 B_361 B_461 B_561 B_661 B_761 B_861;
 % values in Tesla
   B_151 B_251 B_351 B_451 B_551 B_651 B_751 B_851;
   B_141 B_241 B_341 B_441 B_541 B_641 B_741 B_841;
   B_131 B_231 B_331 B_431 B_531 B_631 B_731 B_831;
   B_121 B_221 B_321 B_421 B_521 B_621 B_721 B_821;
   B_111 B_211 B_311 B_411 B_511 B_611 B_711 B_811];
figure(1)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_1);
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
figure(2)
C=contour3(X,Y,B_level_1,10);
clabel(C)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
```

```
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
figure(3)
[DB]=gradient(B_level_1,.5,.5);
D=contour3(X,Y,B_level_1);
clabel(D)
hold on
quiver(B_level_1,DB)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT GROUND LEVEL')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load i1j1z2.txt; load i2j1z2.txt; load i3j1z2.txt; load i4j1z2.txt; load i5j1z2.txt; load i6j1z2.txt;
load i7j1z2.txt; load i8j1z2.txt
load i1j2z2.txt; load i2j2z2.txt; load i3j2z2.txt; load i4j2z2.txt; load i5j2z2.txt; load i6j2z2.txt;
load i7j2z2.txt; load i8j2z2.txt
load i1j3z2.txt; load i2j3z2.txt; load i3j3z2.txt; load i4j3z2.txt; load i5j3z2.txt; load i6j3z2.txt;
load i7j3z2.txt; load i8j3z2.txt
load i1j4z2.txt; load i2j4z2.txt; load i3j4z2.txt; load i4j4z2.txt; load i5j4z2.txt; load i6j4z2.txt;
load i7j4z2.txt; load i8j4z2.txt
load i1j5z2.txt; load i2j5z2.txt; load i3j5z2.txt; load i4j5z2.txt; load i5j5z2.txt; load i6j5z2.txt;
load i7j5z2.txt; load i8j5z2.txt
load i1j6z2.txt; load i2j6z2.txt; load i3j6z2.txt; load i4j6z2.txt; load i5j6z2.txt; load i6j6z2.txt;
load i7j6z2.txt; load i8j6z2.txt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% B field calculations for i1j1z2 through i8j1z2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%n1=[mean(i1j1z2(1:
180,1:3)) mean(i2j1z2(1:180,1:3)) mean(i3j1z2(1:180,1:3)) mean(i4j1z2(1:180,1:3))
mean(i5j1z2(1:180,1:3)) mean(i6j1z2(1:180,1:3)) mean(i7j1z2(1:180,1:3))
mean(i8j1z2(1:180,1:3));
  mean(i1j2z2(1:180,1:3)) mean(i2j2z2(1:180,1:3)) mean(i3j2z2(1:180,1:3))
mean(i4j2z2(1:180,1:3)) mean(i5j2z2(1:180,1:3)) mean(i6j2z2(1:180,1:3))
mean(i7j2z2(1:180,1:3)) mean(i8j2z2(1:180,1:3));
  mean(i1j3z2(1:180,1:3)) mean(i2j3z2(1:180,1:3)) mean(i3j3z2(1:180,1:3))
mean(i4j3z2(1:180,1:3)) mean(i5j3z2(1:180,1:3)) mean(i6j3z2(1:180,1:3))
mean(i8j3z2(1:180,1:3)) mean(i8j3z2(1:180,1:3));
  mean(i1j4z2(1:180,1:3)) mean(i2j4z2(1:180,1:3)) mean(i3j4z2(1:180,1:3))
mean(i4j4z2(1:180,1:3)) mean(i5j4z2(1:180,1:3)) mean(i6j4z2(1:180,1:3))
mean(i8j4z2(1:180,1:3)) mean(i8j4z2(1:180,1:3));
  mean(i1j5z2(1:180,1:3)) mean(i2j5z2(1:180,1:3)) mean(i3j5z2(1:180,1:3))
mean(i4j5z2(1:180,1:3)) mean(i5j5z2(1:180,1:3)) mean(i6j5z2(1:180,1:3))
mean(i8j5z2(1:180,1:3)) mean(i8j5z2(1:180,1:3));
  mean(i1j6z2(1:180,1:3)) mean(i2j6z2(1:180,1:3)) mean(i3j6z2(1:180,1:3))
mean(i4j6z2(1:180,1:3)) mean(i5j6z2(1:180,1:3)) mean(i6j6z2(1:180,1:3))
mean(i8j6z2(1:180,1:3)) mean(i8j6z2(1:180,1:3))];
```

```
B_112=sqrt(n1(1,1)^2+n1(1,2)^2+n1(1,3))/(15000*10000);
B_212=sqrt(n1(1,4)^2+n1(1,5)^2+n1(1,6))/(15000*10000);
B_312=sqrt(n1(1,7)^2+n1(1,8)^2+n1(1,9))/(15000*10000);
B_412=sqrt(n1(1,10)^2+n1(1,11)^2+n1(1,12))/(15000*10000);
B_512=sqrt(n1(1,13)^2+n1(1,14)^2+n1(1,15))/(15000*10000);
B_612=sqrt(n1(1,16)^2+n1(1,17)^2+n1(1,18))/(15000*10000);
B_712=sqrt(n1(1,19)^2+n1(1,20)^2+n1(1,21))/(15000*10000);
B_812=sqrt(n1(1,22)^2+n1(1,23)^2+n1(1,24))/(15000*10000);

B_122=sqrt(n1(2,1)^2+n1(2,2)^2+n1(2,3))/(15000*10000);
B_222=sqrt(n1(2,4)^2+n1(2,5)^2+n1(2,6))/(15000*10000);
B_322=sqrt(n1(2,7)^2+n1(2,8)^2+n1(2,9))/(15000*10000);
B_422=sqrt(n1(2,10)^2+n1(2,11)^2+n1(2,12))/(15000*10000);
B_522=sqrt(n1(2,13)^2+n1(2,14)^2+n1(2,15))/(15000*10000);
B_622=sqrt(n1(2,16)^2+n1(2,17)^2+n1(2,18))/(15000*10000);
B_722=sqrt(n1(2,19)^2+n1(2,20)^2+n1(2,21))/(15000*10000);
B_822=sqrt(n1(2,22)^2+n1(2,23)^2+n1(2,24))/(15000*10000);

B_132=sqrt(n1(3,1)^2+n1(3,2)^2+n1(3,3))/(15000*10000);
B_232=sqrt(n1(3,4)^2+n1(3,5)^2+n1(3,6))/(15000*10000);
B_332=sqrt(n1(3,7)^2+n1(3,8)^2+n1(3,9))/(15000*10000);
B_432=sqrt(n1(3,10)^2+n1(3,11)^2+n1(3,12))/(15000*10000);
B_532=sqrt(n1(3,13)^2+n1(3,14)^2+n1(3,15))/(15000*10000);
B_632=sqrt(n1(3,16)^2+n1(3,17)^2+n1(3,18))/(15000*10000);
B_732=sqrt(n1(3,19)^2+n1(3,20)^2+n1(3,21))/(15000*10000);
B_832=sqrt(n1(3,22)^2+n1(3,23)^2+n1(3,24))/(15000*10000);

B_142=sqrt(n1(4,1)^2+n1(4,2)^2+n1(4,3))/(15000*10000);
B_242=sqrt(n1(4,4)^2+n1(4,5)^2+n1(4,6))/(15000*10000);
B_342=sqrt(n1(4,7)^2+n1(4,8)^2+n1(4,9))/(15000*10000);
B_442=sqrt(n1(4,10)^2+n1(4,11)^2+n1(4,12))/(15000*10000);
B_542=sqrt(n1(4,13)^2+n1(4,14)^2+n1(4,15))/(15000*10000);
B_642=sqrt(n1(4,16)^2+n1(4,17)^2+n1(4,18))/(15000*10000);
B_742=sqrt(n1(4,19)^2+n1(4,20)^2+n1(4,21))/(15000*10000);
B_842=sqrt(n1(4,22)^2+n1(4,23)^2+n1(4,24))/(15000*10000);

B_152=sqrt(n1(5,1)^2+n1(5,2)^2+n1(5,3))/(15000*10000);
B_252=sqrt(n1(5,4)^2+n1(5,5)^2+n1(5,6))/(15000*10000);
B_352=sqrt(n1(5,7)^2+n1(5,8)^2+n1(5,9))/(15000*10000);
B_452=sqrt(n1(5,10)^2+n1(5,11)^2+n1(5,12))/(15000*10000);
B_552=sqrt(n1(5,13)^2+n1(5,14)^2+n1(5,15))/(15000*10000);
B_652=sqrt(n1(5,16)^2+n1(5,17)^2+n1(5,18))/(15000*10000);
B_752=sqrt(n1(5,19)^2+n1(5,20)^2+n1(5,21))/(15000*10000);
B_852=sqrt(n1(5,22)^2+n1(5,23)^2+n1(5,24))/(15000*10000);
```

```
B_162=sqrt(n1(6,1)^2+n1(6,2)^2+n1(6,3))/(15000*10000);
B_262=sqrt(n1(6,4)^2+n1(6,5)^2+n1(6,6))/(15000*10000);
B_362=sqrt(n1(6,7)^2+n1(6,8)^2+n1(6,9))/(15000*10000);
B_462=sqrt(n1(6,10)^2+n1(6,11)^2+n1(6,12))/(15000*10000);
B_562=sqrt(n1(6,13)^2+n1(6,14)^2+n1(6,15))/(15000*10000);
B_662=sqrt(n1(6,16)^2+n1(6,17)^2+n1(6,18))/(15000*10000);
B_762=sqrt(n1(6,19)^2+n1(6,20)^2+n1(6,21))/(15000*10000);
B_862=sqrt(n1(6,22)^2+n1(6,23)^2+n1(6,24))/(15000*10000);

B_level_2=[B_162 B_262 B_362 B_462 B_562 B_662 B_762 B_862;
% values in Tesla
        B_152 B_252 B_352 B_452 B_552 B_652 B_752 B_852;
        B_142 B_242 B_342 B_442 B_542 B_642 B_742 B_842;
        B_132 B_232 B_332 B_432 B_532 B_632 B_732 B_832;
        B_122 B_222 B_322 B_422 B_522 B_622 B_722 B_822;
        B_112 B_212 B_312 B_412 B_512 B_612 B_712 B_812];
figure(4)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_2)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')
figure(5)
C2=contour3(X,Y,B_level_2,10);
clabel(C2)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')
figure(6)
[DB2]=gradient(B_level_2,.5,.5);
D2=contour3(X,Y,B_level_2);
clabel(D2)
hold on
quiver(B_level_2,DB2)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 977 mm ABOVE
FLOOR LEVEL')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%B field calculations for i1j1z3 through i8j1z3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load i1j1z3.txt; load i2j1z3.txt; load i3j1z3.txt; load i4j1z3.txt; load i5j1z3.txt; load i6j1z3.txt;
load i7j1z3.txt; load i8j1z3.txt
```

```
load i1j2z3.txt; load i2j2z3.txt; load i3j2z3.txt; load i4j2z3.txt; load i5j2z3.txt; load i6j2z3.txt;
load i7j2z3.txt; load i8j2z3.txt
load i1j3z3.txt; load i2j3z3.txt; load i3j3z3.txt; load i4j3z3.txt; load i5j3z3.txt; load i6j3z3.txt;
load i7j3z3.txt; load i8j3z3.txt
load i1j4z3.txt; load i2j4z3.txt; load i3j4z3.txt; load i4j4z3.txt; load i5j4z3.txt; load i6j4z3.txt;
load i7j4z3.txt; load i8j4z3.txt
load i1j5z3.txt; load i2j5z3.txt; load i3j5z3.txt; load i4j5z3.txt; load i5j5z3.txt; load i6j5z3.txt;
load i7j5z3.txt; load i8j5z3.txt
load i1j6z3.txt; load i2j6z3.txt; load i3j6z3.txt; load i4j6z3.txt; load i5j6z3.txt; load i6j6z3.txt;
load i7j6z3.txt; load i8j6z3.txt

n3=[mean(i1j1z3(1:180,1:3)) mean(i2j1z3(1:180,1:3)) mean(i3j1z3(1:180,1:3))
mean(i4j1z3(1:180,1:3)) mean(i5j1z3(1:180,1:3)) mean(i6j1z3(1:180,1:3))
mean(i7j1z3(1:180,1:3)) mean(i8j1z3(1:180,1:3));
  mean(i1j2z3(1:180,1:3)) mean(i2j2z3(1:180,1:3)) mean(i3j2z3(1:180,1:3))
mean(i4j2z3(1:180,1:3)) mean(i5j2z3(1:180,1:3)) mean(i6j2z3(1:180,1:3))
mean(i7j2z3(1:180,1:3)) mean(i8j2z3(1:160,1:3));
  mean(i1j3z3(1:180,1:3)) mean(i2j3z3(1:180,1:3)) mean(i3j3z3(1:180,1:3))
mean(i4j3z3(1:180,1:3)) mean(i5j3z3(1:170,1:3)) mean(i6j3z3(1:180,1:3))
mean(i8j3z3(1:180,1:3)) mean(i8j3z3(1:180,1:3));
  mean(i1j4z3(1:180,1:3)) mean(i2j4z3(1:180,1:3)) mean(i3j4z3(1:180,1:3))
mean(i4j4z3(1:180,1:3)) mean(i5j4z3(1:180,1:3)) mean(i6j4z3(1:180,1:3))
mean(i8j4z3(1:180,1:3)) mean(i8j4z3(1:180,1:3));
  mean(i1j5z3(1:180,1:3)) mean(i2j5z3(1:180,1:3)) mean(i3j5z3(1:180,1:3))
mean(i4j5z3(1:180,1:3)) mean(i5j5z3(1:180,1:3)) mean(i6j5z3(1:180,1:3))
mean(i8j5z3(1:180,1:3)) mean(i8j5z3(1:180,1:3));
  mean(i1j6z3(1:180,1:3)) mean(i2j6z3(1:180,1:3)) mean(i3j6z3(1:180,1:3))
mean(i4j6z3(1:180,1:3)) mean(i5j6z3(1:180,1:3)) mean(i6j6z3(1:180,1:3))
mean(i8j6z3(1:180,1:3)) mean(i8j6z3(1:180,1:3))];

B_113=sqrt(n3(1,1)^2+n3(1,2)^2+n3(1,3))/(15000*10000);
B_213=sqrt(n3(1,4)^2+n3(1,5)^2+n3(1,6))/(15000*10000);
B_313=sqrt(n3(1,7)^2+n3(1,8)^2+n3(1,9))/(15000*10000);
B_413=sqrt(n3(1,10)^2+n3(1,11)^2+n3(1,12))/(15000*10000);
B_513=sqrt(n3(1,13)^2+n3(1,14)^2+n3(1,15))/(15000*10000);
B_613=sqrt(n3(1,16)^2+n3(1,17)^2+n3(1,18))/(15000*10000);
B_713=sqrt(n3(1,19)^2+n3(1,20)^2+n3(1,21))/(15000*10000);
B_813=sqrt(n3(1,22)^2+n3(1,23)^2+n3(1,24))/(15000*10000);

B_123=sqrt(n3(2,1)^2+n3(2,2)^2+n3(2,3))/(15000*10000);
B_223=sqrt(n3(2,4)^2+n3(2,5)^2+n3(2,6))/(15000*10000);
B_323=sqrt(n3(2,7)^2+n3(2,8)^2+n3(2,9))/(15000*10000);
B_423=sqrt(n3(2,10)^2+n3(2,11)^2+n3(2,12))/(15000*10000);
B_523=sqrt(n3(2,13)^2+n3(2,14)^2+n3(2,15))/(15000*10000);
B_623=sqrt(n3(2,16)^2+n3(2,17)^2+n3(2,18))/(15000*10000);
```

```
B_723=sqrt(n3(2,19)^2+n3(2,20)^2+n3(2,21))/(15000*10000);
B_823=sqrt(n3(2,22)^2+n3(2,23)^2+n3(2,24))/(15000*10000);

B_133=sqrt(n3(3,1)^2+n3(3,2)^2+n3(3,3))/(15000*10000);
B_233=sqrt(n3(3,4)^2+n3(3,5)^2+n3(3,6))/(15000*10000);
B_333=sqrt(n3(3,7)^2+n3(3,8)^2+n3(3,9))/(15000*10000);
B_433=sqrt(n3(3,10)^2+n3(3,11)^2+n3(3,12))/(15000*10000);
B_533=sqrt(n3(3,13)^2+n3(3,14)^2+n3(3,15))/(15000*10000);
B_633=sqrt(n3(3,16)^2+n3(3,17)^2+n3(3,18))/(15000*10000);
B_733=sqrt(n3(3,19)^2+n3(3,20)^2+n3(3,21))/(15000*10000);
B_833=sqrt(n3(3,22)^2+n3(3,23)^2+n3(3,24))/(15000*10000);

B_143=sqrt(n3(4,1)^2+n3(4,2)^2+n3(4,3))/(15000*10000);
B_243=sqrt(n3(4,4)^2+n3(4,5)^2+n3(4,6))/(15000*10000);
B_343=sqrt(n3(4,7)^2+n3(4,8)^2+n3(4,9))/(15000*10000);
B_443=sqrt(n3(4,10)^2+n3(4,11)^2+n3(4,12))/(15000*10000);
B_543=sqrt(n3(4,13)^2+n3(4,14)^2+n3(4,15))/(15000*10000);
B_643=sqrt(n3(4,16)^2+n3(4,17)^2+n3(4,18))/(15000*10000);
B_743=sqrt(n3(4,19)^2+n3(4,20)^2+n3(4,21))/(15000*10000);
B_843=sqrt(n3(4,22)^2+n3(4,23)^2+n3(4,24))/(15000*10000);

B_153=sqrt(n3(5,1)^2+n3(5,2)^2+n3(5,3))/(15000*10000);
B_253=sqrt(n3(5,4)^2+n3(5,5)^2+n3(5,6))/(15000*10000);
B_353=sqrt(n3(5,7)^2+n3(5,8)^2+n3(5,9))/(15000*10000);
B_453=sqrt(n3(5,10)^2+n3(5,11)^2+n3(5,12))/(15000*10000);
B_553=sqrt(n3(5,13)^2+n3(5,14)^2+n3(5,15))/(15000*10000);
B_653=sqrt(n3(5,16)^2+n3(5,17)^2+n3(5,18))/(15000*10000);
B_753=sqrt(n3(5,19)^2+n3(5,20)^2+n3(5,21))/(15000*10000);
B_853=sqrt(n3(5,22)^2+n3(5,23)^2+n3(5,24))/(15000*10000);

B_163=sqrt(n3(6,1)^2+n3(6,2)^2+n3(6,3))/(15000*10000);
B_263=sqrt(n3(6,4)^2+n3(6,5)^2+n3(6,6))/(15000*10000);
B_363=sqrt(n3(6,7)^2+n3(6,8)^2+n3(6,9))/(15000*10000);
B_463=sqrt(n3(6,10)^2+n3(6,11)^2+n3(6,12))/(15000*10000);
B_563=sqrt(n3(6,13)^2+n3(6,14)^2+n3(6,15))/(15000*10000);
B_663=sqrt(n3(6,16)^2+n3(6,17)^2+n3(6,18))/(15000*10000);
B_763=sqrt(n3(6,19)^2+n3(6,20)^2+n3(6,21))/(15000*10000);
B_863=sqrt(n3(6,22)^2+n3(6,23)^2+n3(6,24))/(15000*10000);

B_level_3=[B_163 B_263 B_363 B_463 B_563 B_663 B_763 B_863;
% values in Tesla
        B_153 B_253 B_353 B_453 B_553 B_653 B_753 B_853;
        B_143 B_243 B_343 B_443 B_543 B_643 B_743 B_843;
        B_133 B_233 B_333 B_433 B_533 B_633 B_733 B_833;
        B_123 B_223 B_323 B_423 B_523 B_623 B_723 B_823;
```

```
        B_113 B_213 B_313 B_413 B_513 B_613 B_713 B_813];
figure(7)
[X,Y]=meshgrid(x,yy);
surf(X,Y,B_level_3)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')
figure(8)
C3=contour3(X,Y,B_level_3,10);
clabel(C3)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')
figure(9)
[DB3]=gradient(B_level_3,.5,.5);
D3=contour3(X,Y,B_level_3);
clabel(D3)
hold on
quiver(B_level_3,DB3)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1250 mm ABOVE
FLOOR LEVEL')

%B field calculations for i1j1z4 through i8j1z4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load i1j1z4.txt; load i2j1z4.txt; load i3j1z4.txt; load i4j1z4.txt; load i5j1z4.txt; load i6j1z4.txt;
load i7j1z4.txt; load i8j1z4.txt
load i1j2z4.txt; load i2j2z4.txt; load i3j2z4.txt; load i4j2z4.txt; load i5j2z4.txt; load i6j2z4.txt;
load i7j2z4.txt; load i8j2z4.txt
load i1j3z4.txt; load i2j3z4.txt; load i3j3z4.txt; load i4j3z4.txt; load i5j3z4.txt; load i6j3z4.txt;
load i7j3z4.txt; load i8j3z4.txt
load i1j4z4.txt; load i2j4z4.txt; load i3j4z4.txt; load i4j4z4.txt; load i5j4z4.txt; load i6j4z4.txt;
load i7j4z4.txt; load i8j4z4.txt
load i1j5z4.txt; load i2j5z4.txt; load i3j5z4.txt; load i4j5z4.txt; load i5j5z4.txt; load i6j5z4.txt;
load i7j5z4.txt; load i8j5z4.txt
load i1j6z4.txt; load i2j6z4.txt; load i3j6z4.txt; load i4j6z4.txt; load i5j6z4.txt; load i6j6z4.txt;
load i7j6z4.txt; load i8j6z4.txt

n4=[mean(i1j1z4(1:180,1:3)) mean(i2j1z4(1:180,1:3)) mean(i3j1z4(1:180,1:3))
mean(i4j1z4(1:180,1:3)) mean(i5j1z4(1:180,1:3)) mean(i6j1z4(1:180,1:3))
mean(i7j1z4(1:180,1:3)) mean(i8j1z4(1:180,1:3));
   mean(i1j2z4(1:180,1:3)) mean(i2j2z4(1:170,1:3)) mean(i3j2z4(1:180,1:3))
mean(i4j2z4(1:180,1:3)) mean(i5j2z4(1:180,1:3)) mean(i6j2z4(1:180,1:3))
mean(i7j2z4(1:180,1:3)) mean(i8j2z4(1:160,1:3));
```

```
   mean(i1j3z4(1:180,1:3)) mean(i2j3z4(1:180,1:3)) mean(i3j3z4(1:180,1:3))
mean(i4j3z4(1:180,1:3)) mean(i5j3z4(1:170,1:3)) mean(i6j3z4(1:180,1:3))
mean(i8j3z4(1:180,1:3)) mean(i8j3z4(1:180,1:3));
   mean(i1j4z4(1:180,1:3)) mean(i2j4z4(1:180,1:3)) mean(i3j4z4(1:180,1:3))
mean(i4j4z4(1:180,1:3)) mean(i5j4z4(1:180,1:3)) mean(i6j4z4(1:180,1:3))
mean(i8j4z4(1:180,1:3)) mean(i8j4z4(1:180,1:3));
   mean(i1j5z4(1:180,1:3)) mean(i2j5z4(1:180,1:3)) mean(i3j5z4(1:180,1:3))
mean(i4j5z4(1:180,1:3)) mean(i5j5z4(1:180,1:3)) mean(i6j5z4(1:180,1:3))
mean(i8j5z4(1:180,1:3)) mean(i8j5z4(1:180,1:3));
   mean(i1j6z4(1:180,1:3)) mean(i2j6z4(1:180,1:3)) mean(i3j6z4(1:180,1:3))
mean(i4j6z4(1:180,1:3)) mean(i5j6z4(1:180,1:3)) mean(i6j6z4(1:180,1:3))
mean(i8j6z4(1:180,1:3)) mean(i8j6z4(1:180,1:3))];

B_114=sqrt(n4(1,1)^2+n4(1,2)^2+n4(1,3))/(15000*10000);
B_214=sqrt(n4(1,4)^2+n4(1,5)^2+n4(1,6))/(15000*10000);
B_314=sqrt(n4(1,7)^2+n4(1,8)^2+n4(1,9))/(15000*10000);
B_414=sqrt(n4(1,10)^2+n4(1,11)^2+n4(1,12))/(15000*10000);
B_514=sqrt(n4(1,13)^2+n4(1,14)^2+n4(1,15))/(15000*10000);
B_614=sqrt(n4(1,16)^2+n4(1,17)^2+n4(1,18))/(15000*10000);
B_714=sqrt(n4(1,19)^2+n4(1,20)^2+n4(1,21))/(15000*10000);
B_814=sqrt(n4(1,22)^2+n4(1,23)^2+n4(1,24))/(15000*10000);

B_124=sqrt(n4(2,1)^2+n4(2,2)^2+n4(2,3))/(15000*10000);
B_224=sqrt(n4(2,4)^2+n4(2,5)^2+n4(2,6))/(15000*10000);
B_324=sqrt(n4(2,7)^2+n4(2,8)^2+n4(2,9))/(15000*10000);
B_424=sqrt(n4(2,10)^2+n4(2,11)^2+n4(2,12))/(15000*10000);
B_524=sqrt(n4(2,13)^2+n4(2,14)^2+n4(2,15))/(15000*10000);
B_624=sqrt(n4(2,16)^2+n4(2,17)^2+n4(2,18))/(15000*10000);
B_724=sqrt(n4(2,19)^2+n4(2,20)^2+n4(2,21))/(15000*10000);
B_824=sqrt(n4(2,22)^2+n4(2,23)^2+n4(2,24))/(15000*10000);

B_134=sqrt(n4(3,1)^2+n4(3,2)^2+n4(3,3))/(15000*10000);
B_234=sqrt(n4(3,4)^2+n4(3,5)^2+n4(3,6))/(15000*10000);
B_334=sqrt(n4(3,7)^2+n4(3,8)^2+n4(3,9))/(15000*10000);
B_434=sqrt(n4(3,10)^2+n4(3,11)^2+n4(3,12))/(15000*10000);
B_534=sqrt(n4(3,13)^2+n4(3,14)^2+n4(3,15))/(15000*10000);
B_634=sqrt(n4(3,16)^2+n4(3,17)^2+n4(3,18))/(15000*10000);
B_734=sqrt(n4(3,19)^2+n4(3,20)^2+n4(3,21))/(15000*10000);
B_834=sqrt(n4(3,22)^2+n4(3,23)^2+n4(3,24))/(15000*10000);

B_144=sqrt(n4(4,1)^2+n4(4,2)^2+n4(4,3))/(15000*10000);
B_244=sqrt(n4(4,4)^2+n4(4,5)^2+n4(4,6))/(15000*10000);
B_344=sqrt(n4(4,7)^2+n4(4,8)^2+n4(4,9))/(15000*10000);
```

```
B_444=sqrt(n4(4,10)^2+n4(4,11)^2+n4(4,12))/(15000*10000);
B_544=sqrt(n4(4,13)^2+n4(4,14)^2+n4(4,15))/(15000*10000);
B_644=sqrt(n4(4,16)^2+n4(4,17)^2+n4(4,18))/(15000*10000);
B_744=sqrt(n4(4,19)^2+n4(4,20)^2+n4(4,21))/(15000*10000);
B_844=sqrt(n4(4,22)^2+n4(4,23)^2+n4(4,24))/(15000*10000);

B_154=sqrt(n4(5,1)^2+n4(5,2)^2+n4(5,3))/(15000*10000);
B_254=sqrt(n4(5,4)^2+n4(5,5)^2+n4(5,6))/(15000*10000);
B_354=sqrt(n4(5,7)^2+n4(5,8)^2+n4(5,9))/(15000*10000);
B_454=sqrt(n4(5,10)^2+n4(5,11)^2+n4(5,12))/(15000*10000);
B_554=sqrt(n4(5,13)^2+n4(5,14)^2+n4(5,15))/(15000*10000);
B_654=sqrt(n4(5,16)^2+n4(5,17)^2+n4(5,18))/(15000*10000);
B_754=sqrt(n4(5,19)^2+n4(5,20)^2+n4(5,21))/(15000*10000);
B_854=sqrt(n4(5,22)^2+n4(5,23)^2+n4(5,24))/(15000*10000);

B_164=sqrt(n4(6,1)^2+n4(6,2)^2+n4(6,3))/(15000*10000);
B_264=sqrt(n4(6,4)^2+n4(6,5)^2+n4(6,6))/(15000*10000);
B_364=sqrt(n4(6,7)^2+n4(6,8)^2+n4(6,9))/(15000*10000);
B_464=sqrt(n4(6,10)^2+n4(6,11)^2+n4(6,12))/(15000*10000);
B_564=sqrt(n4(6,13)^2+n4(6,14)^2+n4(6,15))/(15000*10000);
B_664=sqrt(n4(6,16)^2+n4(6,17)^2+n4(6,18))/(15000*10000);
B_764=sqrt(n4(6,19)^2+n4(6,20)^2+n4(6,21))/(15000*10000);
B_864=sqrt(n4(6,22)^2+n4(6,23)^2+n4(6,24))/(15000*10000);

B_level_4=[B_164 B_264 B_364 B_464 B_564 B_664 B_764 B_864;
% values in Tesla
        B_154 B_254 B_354 B_454 B_554 B_654 B_754 B_854;
        B_144 B_244 B_344 B_444 B_544 B_644 B_744 B_844;
        B_134 B_234 B_334 B_434 B_534 B_634 B_734 B_834;
        B_124 B_224 B_324 B_424 B_524 B_624 B_724 B_824;
        B_114 B_214 B_314 B_414 B_514 B_614 B_714 B_814];
figure(10)
[X,Y]=meshgrid(x,yy)
surf(X,Y,B_level_4)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
figure(11)
C4=contour3(X,Y,B_level_4,10);
clabel(C4)
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
figure(12)
[DB4]=gradient(B_level_4,.5,.5);
```

```
D4=contour3(X,Y,B_level_4);
clabel(D4)
hold on
quiver(B_level_4,DB4)
hold off
xlabel('X GRID COORDINATE');ylabel('Y GRID COORDINATE');zlabel('TESLA')
title('SURVEY 1: MAGNETIC FIELD MAGNITUDE (TESLA) AT 1517 mm ABOVE
FLOOR LEVEL')
```

APPENDIX I. NPSAT1 AIR–BEARING LABORATORY MAGNETIC FIELD SURVEY NO. 3 MATLAB PROGRAM

This Appendix contains the third MATLAB program written to determine the magnetic field that resides in the SSAG laboratory.

```
% LCDR Eric W. Herbert, USN, 7/19/04
% Air Bearing Magnetic Field Determination: Field Measurement Data 3
% Three sets of meausrement values are used. The measurements were
% taken with the magentometer stationed on the air bearing. The air
% bearing was rotated 360 degrees about the Z axis. Samples taken every
% 15 degrees. Pitched plus and minus % 15 degrees.
% This algorithm's purpose is to load the measured data,
% remove the magnetic north measurement (not required), convert
% the Honeywell HM2300 Magnetometer cooridinate system to the
% cooridinate system laid out in the lab. The magnetometer readings are
% measured in counts. These are converted to Tesla by the equation:
% (counts(1 gauss/15000 counts)(1 tesla/10000 gauss). Additionally, it will
% identify the number of samples as 280.  This is from approx 10 seconds worth of
% measurements. The samples will be averaged and used to calculate the
% magnetic field about the air bearing as it is rotated 360 and pitched 15
% degrees.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all,clf,clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TIER TWO: SECOND TEST
% load measured values
load Incln0_000.txt; load Incln0_030.txt; load Incln0_060.txt; load Incln0_090.txt;
load Incln0_120.txt; load Incln0_150.txt; load Incln0_180.txt; load Incln0_210.txt;
load Incln0_240.txt; load Incln0_270.txt; load Incln0_300.txt; load Incln0_330.txt;

load I30_000.txt; load I30_030.txt; load I30_060.txt; load I30_090.txt;
load I30_120.txt; load I30_150.txt; load I30_180.txt; load I30_210.txt;
load I30_240.txt; load I30_270.txt; load I30_300.txt; load I30_330.txt;

load Ineg30_000.txt; load Ineg30_030.txt; load Ineg30_060.txt; load Ineg30_090.txt;
load Ineg30_120.txt; load Ineg30_150.txt; load Ineg30_180.txt; load Ineg30_210.txt;
load Ineg30_240.txt; load Ineg30_270.txt; load Ineg30_300.txt; load Ineg30_330.txt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m=[mean(Incln0_000(1:280,1:3)); mean(Incln0_030(1:280,1:3));
mean(Incln0_060(1:280,1:3)); mean(Incln0_090(1:280,1:3));
```

```
  mean(Incln0_120(1:280,1:3)); mean(Incln0_150(1:280,1:3)); mean(Incln0_180(1:280,1:3));
mean(Incln0_210(1:280,1:3));
  mean(Incln0_240(1:280,1:3)); mean(Incln0_270(1:280,1:3)); mean(Incln0_300(1:280,1:3));
mean(Incln0_330(1:280,1:3))];
m=m.*1.5e-8;

m_mag=[(sqrt(m(1,1)^2+m(1,2)^2+m(1,3)^2));
      (sqrt(m(2,1)^2+m(2,2)^2+m(2,3)^2));
      (sqrt(m(3,1)^2+m(3,2)^2+m(3,3)^2));
      (sqrt(m(4,1)^2+m(4,2)^2+m(4,3)^2));
      (sqrt(m(5,1)^2+m(5,2)^2+m(5,3)^2));
      (sqrt(m(6,1)^2+m(6,2)^2+m(6,3)^2));
      (sqrt(m(7,1)^2+m(7,2)^2+m(7,3)^2));
      (sqrt(m(8,1)^2+m(8,2)^2+m(8,3)^2));
      (sqrt(m(9,1)^2+m(9,2)^2+m(9,3)^2));
      (sqrt(m(10,1)^2+m(10,2)^2+m(10,3)^2));
      (sqrt(m(11,1)^2+m(11,2)^2+m(11,3)^2));
      (sqrt(m(12,1)^2+m(12,2)^2+m(12,3)^2))];

m1=[mean(I30_000(1:280,1:3)); mean(I30_030(1:280,1:3)); mean(I30_060(1:280,1:3));
mean(I30_090(1:280,1:3));
  mean(I30_120(1:280,1:3)); mean(I30_150(1:280,1:3)); mean(I30_180(1:280,1:3));
mean(I30_210(1:280,1:3));
  mean(I30_240(1:280,1:3)); mean(I30_270(1:280,1:3)); mean(I30_300(1:280,1:3));
mean(I30_330(1:280,1:3))];
m1=m1.*1.5e-8;
m1_mag=[(sqrt(m1(1,1)^2+m1(1,2)^2+m1(1,3)^2));
      (sqrt(m1(2,1)^2+m1(2,2)^2+m1(2,3)^2));
      (sqrt(m1(3,1)^2+m1(3,2)^2+m1(3,3)^2));
      (sqrt(m1(4,1)^2+m1(4,2)^2+m1(4,3)^2));
      (sqrt(m1(5,1)^2+m1(5,2)^2+m1(5,3)^2));
      (sqrt(m1(6,1)^2+m1(6,2)^2+m1(6,3)^2));
      (sqrt(m1(7,1)^2+m1(7,2)^2+m1(7,3)^2));
      (sqrt(m1(8,1)^2+m1(8,2)^2+m1(8,3)^2));
      (sqrt(m1(9,1)^2+m1(9,2)^2+m1(9,3)^2));
      (sqrt(m1(10,1)^2+m1(10,2)^2+m1(10,3)^2));
      (sqrt(m1(11,1)^2+m1(11,2)^2+m1(11,3)^2));
      (sqrt(m1(12,1)^2+m1(12,2)^2+m1(12,3)^2))];

m2=[mean(Ineg30_000(1:280,1:3)); mean(Ineg30_030(1:280,1:3));
mean(Ineg30_060(1:280,1:3)); mean(Ineg30_090(1:280,1:3));
  mean(Ineg30_120(1:280,1:3)); mean(Ineg30_150(1:280,1:3));
mean(Ineg30_180(1:280,1:3)); mean(Ineg30_210(1:280,1:3));
  mean(Ineg30_240(1:280,1:3)); mean(Ineg30_270(1:280,1:3));
mean(Ineg30_300(1:280,1:3)); mean(Ineg30_330(1:280,1:3))];
```

```
m2=m2.*1.5e-8;
m2_mag=[(sqrt(m2(1,1)^2+m2(1,2)^2+m2(1,3)^2));
      (sqrt(m2(2,1)^2+m2(2,2)^2+m2(2,3)^2));
      (sqrt(m2(3,1)^2+m2(3,2)^2+m2(3,3)^2));
      (sqrt(m2(4,1)^2+m2(4,2)^2+m2(4,3)^2));
      (sqrt(m2(5,1)^2+m2(5,2)^2+m2(5,3)^2));
      (sqrt(m2(6,1)^2+m2(6,2)^2+m2(6,3)^2));
      (sqrt(m2(7,1)^2+m2(7,2)^2+m2(7,3)^2));
      (sqrt(m2(8,1)^2+m2(8,2)^2+m2(8,3)^2));
      (sqrt(m2(9,1)^2+m2(9,2)^2+m2(9,3)^2));
      (sqrt(m2(10,1)^2+m2(10,2)^2+m2(10,3)^2));
      (sqrt(m2(11,1)^2+m2(11,2)^2+m2(11,3)^2));
      (sqrt(m2(12,1)^2+m2(12,2)^2+m2(12,3)^2))];


r=.0254*9.5;
r1=.866*9.5*.0254;
r2=r1;

phi=linspace(0,360,12)*pi/180;
x=(r*cos(phi))';
y=(r*sin(phi))';
x1=(r2*cos(phi))';
y1=(r2*sin(phi))';
x2=x1;
y2=y1;
z=zeros(size(phi'));
z1=.5*ones(size(phi'));
z2=-z1;
figure(1)
plot3(x,y,z),grid on, hold on
plot3(x1,y1,z1)
plot3(x2,y2,z2)
quiver3(x,y,z,m_mag(:,1),m(:,2),m(:,3),'r')
quiver3(x1,y1,m1_mag,m1(:,1),m1(:,2),m1(:,3),'b')
quiver3(x2,y2,m2_mag,m2(:,1),m2(:,2),m2(:,3),'g')
xlabel('X'),ylabel('Y'),zlabel('Normalized Magnetic Field Direction')
title('NPSAT1 AIR-BEARING MAGNETIC FIELD DIRECTION: SURVEY 3, TIERS 1,2,3')
figure(2)
plot3(x,y,m_mag,'r'),grid on
xlabel('X'),ylabel('Y'),zlabel('Magnetic Field Magnitude (Tesla)')
title('NPSAT1 AIR-BEARING MAGNETIC FIELD MAGNITUDE; SURYVEY 3,TIER 1')
figure(3)
plot3(x,y,m_mag,'r'),grid on
hold on
```

```
quiver3(x,y,m_mag,m(:,1),m(:,2),m(:,3),'r')
xlabel('X'),ylabel('Y'),zlabel('Normalized Magnetic Field Magnitude')
title('NPSAT1 AIR-BEARING MAGNETIC FIELD MAGNITUDE NORMALIZED VEC-
TOR;  SURVEY 3,TIER 1')
```

APPENDIX J. NPSAT1 AIR–BEARING "HARDWARE-IN-THE-LOOP" DYNAMIC MODEL

This Appendix displays the SIMULINK model and sub–models used during air–bearing *hardware-in-the-loop* expiraments.

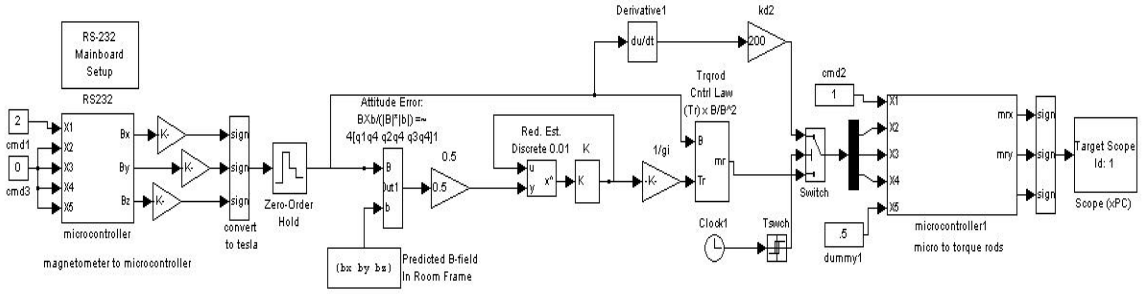


Figure J.1 NPSAT1 "Hardware-in-the-Loop" Simulink Model

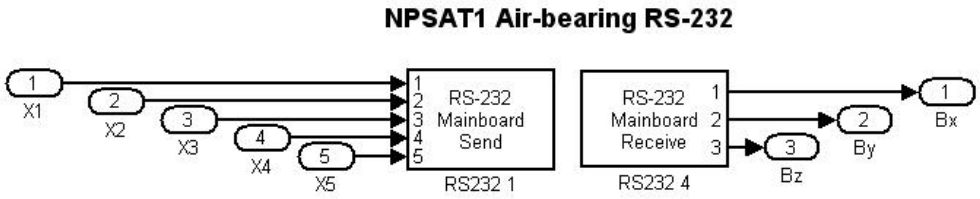
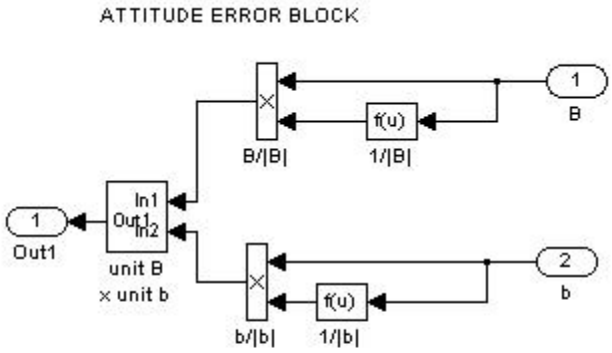
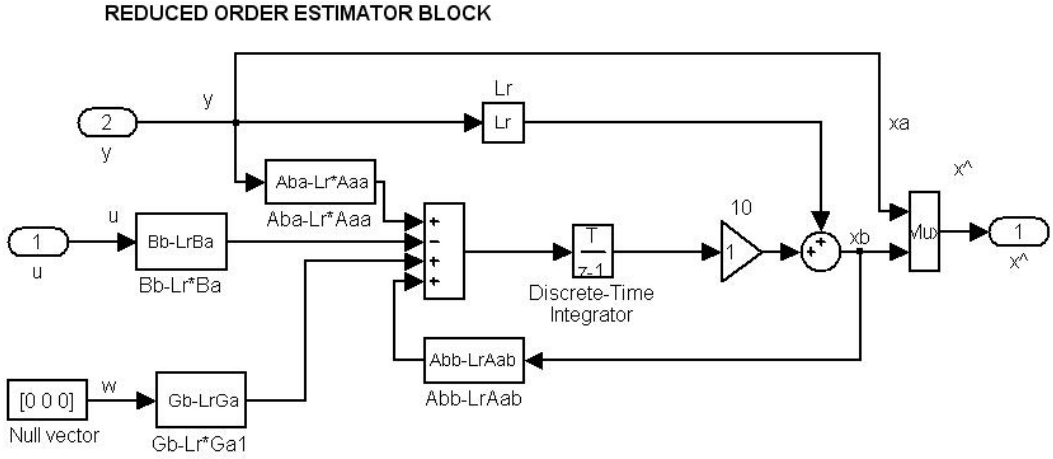


Figure J.2 RS232 Communication Channels



This block represents an approximation to the position part of the quaternion control law. Sidi (Ref 5, pg 156) suggests using e = [q1q4 q2q4 q3q4] as an attitude error vector. The cross product between unit B and unit b is an approximation to this error vector for magnetic magnetic control systems, where: B = magnetometer output and b = predicted magnetic field vector based on emphemeris data and a spherical harmonic model

Figure J.3 NPSAT1 ACS Attitude Error Sub–Model



DEFINITIONS : Given the equations of state xdot = Ax +Bu +Gw and y =Cx +Du, where for this problem x =[phi,theta,psi,phidot,thetadot,psidot]' =:[x1, x2, x3, x4, x5, x6]', The reduced order estimator is used toestimate xb =: [x4, x5, x6]' from the input xa =:[x1,x2,x3]'. For this problem, y=xa (i.e. C =eye(3) and D=0). Similar to the partitioning of x =[xa, xb]' , A=[Aaa Aab; Aba Abb], B=[Ba ; Bb], and G=[Ga ; Gb]. Where A, B, G are defined in the M-file and the corresponding block diagram is shown above (REF 1). The controller gain K is derived using the "lqr" Matlab command. Attempts to derive the reduced estimator gain Lr using the Matlab "place" and "kalman" commands were not very satisfying; but, Lr = 0.2*eye(3) produced satisfactory responses.

Reference 1. "Feedback Control Of Dynamic Systems"; Franklin, Powell, Emami-Naeini; Addison Wesley ,1988 pp 352-356.

Figure J.4 NPSAT1 ACS Reduced Order Estimator Sub–Model

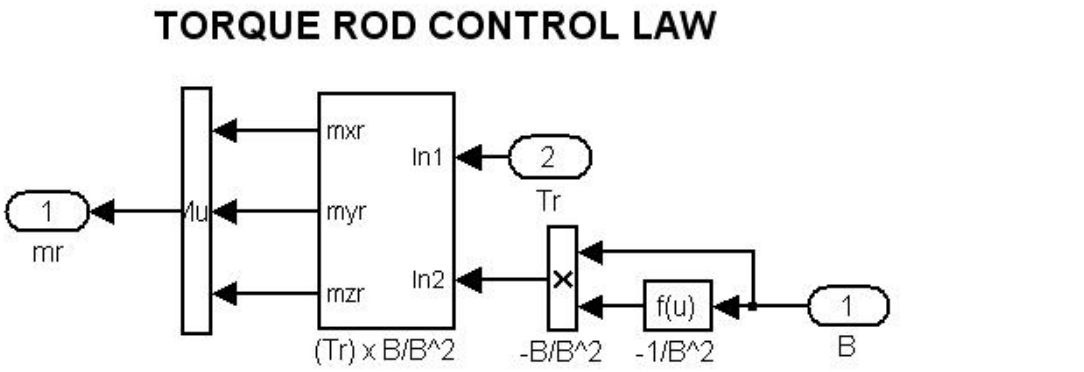


Figure J.5 NPSAT1 ACS Torque Rod Control Law Sub–Model

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX K. NPSAT1 AIR–BEARING "HARDWARE-IN-THE-LOOP" PROGRAM

This Appendix contains the MATLAB program that was drafted to be used with model displayed in Appendix J.

```
% LCDR Eric Herbert, USN, NPS
% NPSAT1 Magnetic ACS Air-Bearing Simulation
% Modeled after NPSAT1 ACS Algorithm
% The algorithm is designed to accomplish two tasks:
% 1) simulate and plot the torque output based on
%    labratory mag field surveys and corrected 2005 IGRF data
% 2) interface the NPSAT1 ACS REO plant to the RS232 I/O device
%    and the XP Target/Host applications
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%clear,
clc
Re=6.37814e6;          % Earth equatorial radius (meters): p. 827 Wertz
mu=3.98601e14;         % gravity constant (m^3/s^2)
h=1;                   % altitude (meters)
ms=50;                 % air bearing table mass (kg)
gr=9.8066;             % gravity constant m/s^2
lv=0.001;              % Table mass~kg, g~m/s^2, lever arm~m
r=Re+h;                %definitions to match Wertz pg 780
a=Re;
wo=0;                  % sqrt(mu/(Re+h)^3); equivalent orbit rate

% Greatest Field components in Room frame
% at theta = 90 degrees inclination from Z axis; phi = 120 degrees
% data collected from lab survey # 3
bx=8.17e-5;
by=5.28e-5;
bz=1.93e-5;
b2=(bx.^2+by.^2+bz.^2);

% Table moments of inertia (MOI)~kg.m^2 as of 1 Aug 2004

Ix=2.743178; Iy=2.642236; Iz=2.007684;
Ixy=-0.003267051; Ixz=0.03089219; Iyz=-0.002887149;
Imoi=[Ix -Ixy -Ixz;-Ixy Iy -Iyz;-Ixz -Iyz Iz]; %MOI matrix
Iinv=inv(Imoi);                               %MOI matrix inverse

pho=.1*3;tho=-.1*3;pso=pi/3;                  %initial Euler angles (phi,theta,psi) ~r
phdo=-0.1;thdo=0.1;psdo=0.01;                 %initial Euler angle rates ~r/s
```

```
phdo=-0.0;thdo=0.0;psdo=0.;

%calculation of initial quaternion (qo) and angular momentum (Ho):

s1=sin(pho/2); s2=sin(tho/2); s3=sin(pso/2);
c1=cos(pho/2); c2=cos(tho/2); c3=cos(pso/2);
q1o=s1*c2*c3-c1*s2*s3;              %Wie pg.321
q2o=c1*s2*c3+s1*c2*s3;
q3o=c1*c2*s3-s1*s2*c3;q4o=c1*c2*c3+s1*s2*s3;
S1=sin(pho);S2=sin(tho);S3=sin(pso);C1=cos(pho);C2=cos(tho);C3=cos(pso);
wxo=phdo-psdo*S2-wo*S3*C2;
wyo=thdo*C1+psdo*C2*S1-wo*(C3*C1+S3*S2*S1);
wzo=psdo*C2*C1-thdo*S1-wo*(S3*S2*C1-C3*S1);
qo=[q1o q2o q3o q4o];  Ho=Imoi*[wxo wyo wzo]';

%Calculation of spherical harmonic magnetic field model (Wertz pp779-783)

% IGRF Epoch 2000 Guassian coefficients ~ n.T   :
g_initial=[-29615 -1728 0 0 0 0 0 0 0 0 0;            %2000 IGRF g coef ref (1)
        -2267 3072 1672 0 0 0 0 0 0 0 0;
        1341 -2290 1253 715 0 0 0 0 0 0 0;
        935 787 251 -405 110 0 0 0 0 0 0;
        -217 351 222 -131 -169 -12 0 0 0 0 0;
        72 68 74 -161 -5 17 -91 0 0 0 0;
        79 -74 0 33 9 7 8 -2 0 0 0;
        25 6 -9 -8 -17 9 7 -8 -7 0 0;
        5 9 3 -8 6 -9 -2 9 -4 -8 0;
        -2 -6 2 -3 0 4 1 2 4 0 -1];

g_correct_factor=5*[14.6 10.7 0 0 0 0 0 0 0 0 0;  % 2005 IGRF g coef correction factor
                -12.4 1.1 -1.1 0 0 0 0 0 0 0 0;          % which is multiplied by scaler  5
                0.7 -5.4 0.9 -7.7 0 0 0 0 0 0 0;         % for each year from 2000
                -1.3 1.6 -7.3 2.9 -3.2 0 0 0 0 0 0;      % same concept for h correction
                0 -0.7 -2.1 -2.8 -0.8 2.5 0 0 0 0 0;
                1 -0.4 0.9 2 -0.6 -0.3 1.2 0 0 0 0;
                -0.4 -0.4 -0.3 1.1 1.1 -0.2 0.6 -0.9 0 0 0;
                -0.3 0.2 -0.3 0.4 -1 0.3 -0.5 -0.7 -0.4 0 0;
                 0 0 0 0 0 0 0 0 0 0 0;
                 0 0 0 0 0 0 0 0 0 0 0];

gg=g_initial+g_correct_factor;                  %IGRF2000 g coefficient corrected to 2005

h_initial=[0 5186 0 0 0 0 0 0 0 0 0;            % 2000 IGRF h coef
        0 -2478 -458 0 0 0 0 0 0 0 0;
        0 -227 296 -492 0 0 0 0 0 0 0;
```

```
        0 272 -232 119 -304 0 0 0 0 0 0;
        0 44 172 -134 -40 107 0 0 0 0 0;
        0 -17 64 65 -61 1 44 0 0 0 0;
        0 -65 -24 6 24 15 -25 -6 0 0 0;
        0 12 -22 8 -21 15 9 -16 -3 0 0;
        0 -20 13 12 -6 -8 9 4 -8 5 0;
        0 1 0 4 5 -6 -1 -3 0 -2 -8];

h_correct_factor=5*[0 -22.5 0 0 0 0 0 0 0 0 0;          % h coefficient correction factor
                0 -20.6 -9.6 0 0 0 0 0 0 0 0;
                0 6 -0.1 -14.2 0 0 0 0 0 0 0;
                0 2.1 1.3 5 0.3 0 0 0 0 0 0;
                0 -0.1 0.6 1.7 1.9 0.1 0 0 0 0 0;
                0 -0.2 -1.4 0 -0.8 0 0.9 0 0 0 0;
                0 1.1 0 0.3 -0.1 -0.6 -0.7 0.2 0 0 0;
                0 0.1 0 0 0.3 0.6 -0.4 0.3 0.7 0 0;
                0 0 0 0 0 0 0 0 0 0 0;
                0 0 0 0 0 0 0 0 0 0 0];

 hh=h_initial+h_correct_factor;     % IGRF2000 h coeffiecient corrected to 2005

Kme=(a/r)^3*sqrt(gg(1,1)^2+gg(1,2)^2+hh(1,2)^2)*1e-9;   %dipole strength~Tesla
sat=30;  Tswitch=10; ka=[1 1 1];

%Calc of state space A,B,G matrices follows (xdot=Ax+Bu+Gw,w=disturbance)

A=[0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1;-ms*gr*lv/Ix 0 0 0 0 0;...
   0 -ms*gr*lv/Iy 0 0 0 0;0 0 0 0 0 0];
B=Kme*[0 0 0;0 0 0;0 0 0;ka(1)/Ix 0 0;0 ka(2)/Iy 0;0 0 ka(3)/Iz];
C=[eye(3),zeros(3)];
G=[0 0 0;0 0 0;0 0 0;1/Ix 0 0;0 1/Iy 0;0 0 1/Iz];
wr=sqrt(ms*gr*lv/Ix);
wp=sqrt(ms*gr*lv/Iy);
Qx=1*eye(6);
Ru=1e-7*diag([1 1 2]);        %[4 1 1.5]@20% modulation
[Klqr,S,e]=lqr(A,B,Qx,Ru);
Kp=Klqr(1:3,1:3);
Kd=Klqr(1:3,4:6);
pe=6*[-abs(e(1)) -abs(e(3))  -abs(e(5))];
K=Klqr;

%partitioning of A,B,G matrices required for reduced order estimator:
Aaa=A(1:3,1:3);Aab=A(1:3,4:6);Aba=A(4:6,1:3);Abb=A(4:6,4:6);
Ba=B(1:3,1:3);Bb=B(4:6,1:3);Ka=K(1:3,1:3);Kb=K(1:3,4:6);
Ga=G(1:3,1:3);Gb=G(4:6,1:3);
```

```
Lr=(place(Abb',Aab',pe))';
gx=(by^2+bz^2)/b2; gy=(bx^2+bz^2)/b2;  gz=(bx^2+by^2)/b2;
%sim('airbearing_simulation')
sim('acs_airbearing_static_test_model_5aug04')
%sim('ACS_Airbearing_Static_Test_Model_28JUL04')
figure(1)
plot(tout,Tx,'r',tout,Ty,'b',tout,Tz,'g'),grid on
title('NPSAT1 Magnetic ACS 3-Axis Torque Generation')
xlabel('Seconds'), ylabel('Torque (Newton-Meters)')
%axis([0 .8 -3.5e12 1.5e12])
LEGEND('Tx','Ty','Tz')
tau=sqrt(Tx.^2+Ty.^2+Tz.^2);
figure(2)
plot(tout,tau),grid on
title('NPSAT1 Magnetic ACS 3-Axis Torque Magnitude')
xlabel('Seconds'),ylabel('Torque (Newton-Meters)')
%axis([0 1 1e10 5e12])

%Simulink RS232 set up parameters
RS232.SendData = '%d,%d,%d,%d,%d\n';    % d#1=command ; d#2-d#4=tx,ty,tz ; d#5=time
duration
RS232.RecData = '%d,%d,%d\r';                % d#1-3=magnetometer readings, bx,by,bz
RS232.InputPorts = [1 2 3 4 5];        % identification of serial port block parameters
RS232.OutputPorts = [1 2 3];           % identification of serial port block parameters
RS232.Timeout = 0.005;                 % time out value must be less than 50 millisec-
onds?(maybe)
RS232.EOM = 1;                         % end of message command

% figure(3)
% plot(tout,phi,'b',tout,theta,'g',tout,psi,'r'), grid on
% title('Euler Angle Decay')
% xlabel('Seconds'), ylabel('Angle (degree)')
% %axis([0 1000 -0.4 1])
% legend('Phi','Theta','Psi')
% figure(4)
% plot(tout,mxr,'g',tout,myr,'r',tout,mzr,'b'),grid on
% title('NPSAT1 Magnetic ACS Dipole Moments')
% xlabel('Seconds'),ylabel('Amp meter^2')
% %axis([0 800 -1000 1000])
% legend('mxr','mry','mrz')
% figure(5)
% plot(tout,Torque(:,1),'b',tout,Torque(:,2),'r',tout,Torque(:,3),'g'),grid on
% title('NPSAT1 Solenoid Torque'),xlabel('Seconds'),ylabel('Newton meters')
% %axis([0 800 -2500 2500])
% legend('Torque-X','Torque-Y','Torque-Z')
```

APPENDIX L. NPSAT1 AIR–BEARING SOLENOID FM-1 MAGNETIC FIELD AND GENERATED TORQUE PROGRAMS

This Appendix contains the MATLAB programs drafted to analyze data collected while measuring the magnetic fields of the MICROCOSM magnetic torque rods.

```
% LCDR Eric W. Herbert, USN, Naval Post Graduate School, Monterey, CA
% 18 August 2004, X Torque Rod Magnetic Field and Torque Calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%clear
all, clc
fs=20;                % sampling frequency
ts=1/fs;              % sampling period
r=13.5/1000;          % rod radius (meters)
A=pi*r^2;             % Area enclosed by each turn in coil
mu_M=33;              % rod dipole moment(amp meters^2) at 145 mA
max_i=145e-3;         % max current through solenoid (amps)
N=mu_M/(max_i*pi*r^2);  % # coils on rod
% current at bit levels 255,128,64,32,16,8,4,2,1
I=[140e-3 70e-3 35e-3 17.5e-3 8.75e-3 4.375e-5 2.1875e-3 1.09375e-3 .546875e-3];
Bits=[255:-1:0];
I_step=linspace(140e-3,.546875e-3,256);
mu=N*A.*I;        % calculated dipole moments as I decreases

% load measured data
X255=load ('XMFLD255.txt'); X128=load('XMFLD128.txt');X64=load('XMFLD64.txt');
X32=load('XMFLD32.txt'); X16=load('XMFLD16.txt'); X8=load('XMFLD8.txt');
X4=load('XMFLD4.txt'); X2=load('XMFLD2.txt'); X1=load('XMFLD1.txt');

% extract magnetic north bearing from measured data
X255=X255(:,1:3); X128=X128(:,1:3); X64=X64(:,1:3);
X32=X32(:,1:3); X16=X16(:,1:3); X8=X8(:,1:3);
X4=X4(:,1:3); X2=X2(:,1:3); X1=X1(:,1:3);

% external magnetic field (Tesla)
B=sqrt((X255(1,1)).^2+(X255(1,2)).^2+(X255(1,3)).^2);
Bfld=B.*(1.5e-8);

% convert induced magnetic field (Tesla)
% and Torque Measurements(Newton meters)

for i=1:size(X255)
   X255A(i,:)=X255(1,:)-X255(i,:);
   X255B=X255A.*(1.5e-8);
```

```
    X255C=X255.*(1.5e-8);
    B255(i)=sqrt((X255B(i,1)).^2+(X255B(i,2)).^2+(X255B(i,3)).^2);
    T255=B255.*mu(1);
    t=[0:.05:20.5];
end
for i=1:size(X128)
    X128A(i,1:3)=X128(1,1:3)-X128(i,1:3);
    X128B=X128A.*(1.5e-8);
    X128C=X128.*(1.5e-8);
    B128(i)=sqrt((X128B(i,1)).^2+(X128B(i,2)).^2+(X128B(i,3)).^2);
    T128=B128.*mu(2);
    t1=[0:ts:14.3];
end
for i=1:size(X64)
    X64A(i,1:3)=X64(1,1:3)-X64(i,1:3);
    X64B=X64A.*(1.5e-8);
    X64C=X64.*(1.5e-8);
    B64(i)=sqrt((X64B(i,1)).^2+(X64B(i,2)).^2+(X64B(i,3)).^2);
    T64=B64.*mu(3);
    t2=[0:ts:12.65];
end
for i=1:size(X32)
    X32A(i,1:3)=X32(1,1:3)-X32(i,1:3);
    X32B=X32A.*(1.5e-8);
    X32C=X32.*(1.5e-8);
    B32(i)=sqrt((X32B(i,1)).^2+(X32B(i,2)).^2+(X32B(i,3)).^2);
    T32=B32.*mu(4);
    t3=[0:ts:12.9];
end

for i=1:size(X16)
    X16A(i,1:3)=X16(1,1:3)-X16(i,1:3);
    X16B=X16A.*(1.5e-8);
    X16C=X16.*(1.5e-8);
    B16(i)=sqrt((X16B(i,1)).^2+(X16B(i,2)).^2+(X16B(i,3)).^2);
    T16=B16.*mu(5);
    t4=[0:ts:11.3];
end

for i=1:size(X8)
    X8A(i,1:3)=X8(1,1:3)-X8(i,1:3);
    X8B=X8A.*(1.5e-8);
    X8C=X8.*(1.5e-8);
    B8(i)=sqrt((X8B(i,1)).^2+(X8B(i,2)).^2+(X8B(i,3)).^2);
    T8=B8.*mu(6);
```

```
   t5=[0:ts:9.7];
end
for i=1:size(X4)
   X4A(i,1:3)=X4(1,1:3)-X4(i,1:3);
   X4B=X4A.*(1.5e-8);
   X4C=X4.*(1.5e-8);
   B4(i)=sqrt((X4B(i,1)).^2+(X4B(i,2)).^2+(X4B(i,3)).^2);
   T4=B4.*mu(7);
   t6=[0:ts:10.55];
end
for i=1:size(X2)
   X2A(i,1:3)=X2(1,1:3)-X2(i,1:3);
   X2B=X2A.*(1.5e-8);
   X2C=X2.*(1.5e-8);
   B2(i)=sqrt((X2B(i,1)).^2+(X2B(i,2)).^2+(X2B(i,3)).^2);
   T2=B2.*mu(8);
   t7=[0:ts:11.7];
end
for i=1:size(X1)
   X1A(i,1:3)=X1(1,1:3)-X1(i,1:3);
   X1B=X1A.*(1.5e-8);
   X1C=X1.*(1.5e-8);
   B1(i)=sqrt((X1B(i,1)).^2+(X1B(i,2)).^2+(X1B(i,3)).^2);
   T1=B1.*mu(9);
   t8=[0:ts:8.3];
end
% Capture Data
figure(1)
subplot(2,2,1)
plot(t,B255), axis([8 15 0 5e-4]),grid on
title('X T-Rod Mag Field (255)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t,T255), axis([8 14 0 0.018]),grid on
title('X T-Rod Torque (255)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t1,B128),axis([4 10 0 5e-4]),grid on
title('X T-Rod Mag Field (128)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t1,T128), axis([4 10 0 8e-3]),grid on
title('X T-Rod Torque (128)')
xlabel('Seconds'),ylabel('Newton*Meters')
```

```
figure(2)
subplot(2,2,1)
plot(t2,B64),axis([4 10 0 5e-4]) ,grid on
title('X T-Rod Mag Field (64)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t2,T64), axis([4 10 0 8e-3]),grid on
title('X T-Rod Torque (64)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t3,B32),axis([3 9 0 7.5e-4]) ,grid on
title('X T-Rod Mag Field (32)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t3,T32),grid on,axis([2 8 0 3e-3])
title('X T-Rod Torque (32)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(3)
subplot(2,2,1)
plot(t4,B16), axis([3 9 0 4e-4]),grid on
title('X T-Rod Mag Field (16)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t4,T16),grid on,axis([2 7 0 8e-4])
title('X T-Rod Torque (16)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t5,B8),axis([2 8 0 3e-4]),grid on
title('X T-Rod Mag Field (8)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t5,T8),grid on,axis([2 7 0 2.5e-6])
title('X T-Rod Torque (8)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(4)
subplot(2,2,1)
plot(t6,B4), axis([3 9 0 1.1e-4]),grid on
title('X T-Rod Mag Field (4)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t6,T4),grid on, axis([3 8 0 5.5e-5])
title('X T-Rod Torque (4)')
xlabel('Seconds'),ylabel('Newton*Meters')
```

```
subplot(2,2,3)
plot(t7,B2),axis([2 7 0 1e-4]),grid on
title('X T-Rod Mag Field (2)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t7,T2),grid on, axis([2 7 0 1.3e-5])
title('X T-Rod Torque (2)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(5)
subplot(2,2,1)
plot(t8,B1), axis([2 7 0 2.5e-5]),grid on
title('X T-Rod Mag Field (1)')
xlabel('Seconds'),ylabel('Tesla')
subplot (2,2,2)
plot(t8,T1),grid on, axis([2 7 0 2.5e-6])
title('X T-Rod Torque (1)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,1,2)
semilogy(t,T255,t1,T128,t2,T64,t3,T32,t4,T16,t5,T8,t6,T4,t7,T2,t8,T1)
grid on, axis([0 14 10e-10 1])
title('FM-1 TORQUE RANGE')
xlabel('Seconds'), ylabel('Newton Meters')
legend('255','128','64','32','16','8','4','2','1')

% LCDR Eric W. Herbert, USN, Naval Post Graduate School, Monterey, CA
% 18 August 2004, Y Torque Rod Magnetic Field and Torque Calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
fs=20;              % sampling frequency
ts=1/fs;            % sampling period
r=13.5/1000;        % rod radius (meters)
A=pi*r^2;           % Area enclosed by each turn in coil
mu_M=33;            % rod dipole moment(amp meters^2) at 145 mA
max_i=145e-3;       % max current through solenoid (amps)
N=mu_M/(max_i*pi*r^2);    % # coils on rod
% current at bit levels 255,128,64,32,16,8,4,2,1
I=[140e-3 70e-3 35e-3 17.5e-3 8.75e-3 4.375e-5 2.1875e-3 1.09375e-3 .546875e-3];
Bits=[255:-1:0];
I_step=linspace(140e-3,.546875e-3,256);
mu=N*A.*I;        % calculated dipole moments as I decreases

% load measured data
Y255=load ('YMFLD255.txt'); Y128=load('YMFLD128.txt');Y64=load('YMFLD64.txt');
Y32=load('YMFLD32.txt'); Y16=load('YMFLD16.txt'); Y8=load('YMFLD8.txt');
```

```
Y4=load('YMFLD4.txt'); Y2=load('YMFLD2'); Y1=load('YMFLD1.txt');

% extract magnetic north bearing from measured data
Y255=Y255(:,1:3); Y128=Y128(:,1:3); Y64=Y64(:,1:3);
Y32=Y32(:,1:3); Y16=Y16(:,1:3); Y8=Y8(:,1:3);
Y4=Y4(:,1:3); Y2=Y2(:,1:3); Y1=Y1(:,1:3);

% external magnetic field (Tesla)
B=sqrt((Y255(1,1)).^2+(Y255(1,2)).^2+(Y255(1,3)).^2);
Bfld=B.*(1.5e-8);

% convert induced magnetic field (Tesla)
% and Torque measurement (Newton meters)
for i=1:size(Y255)
   Y255A(i,:)=Y255(1,:)-Y255(i,:);
   Y255B=Y255A.*(1.5e-8);
   Y255C=Y255.*(1.5e-8);
   B255(i)=sqrt((Y255B(i,1)).^2+(Y255B(i,2)).^2+(Y255B(i,3)).^2);
   T255=B255.*mu(1);
   t=[0:.05:21.65];
end
for i=1:size(Y128)
   Y128A(i,1:3)=Y128(1,1:3)-Y128(i,1:3);
   Y128B=Y128A.*(1.5e-8);
   Y128C=Y128.*(1.5e-8);
   B128(i)=sqrt((Y128B(i,1)).^2+(Y128B(i,2)).^2+(Y128B(i,3)).^2);
   T128=B128.*mu(2);
   t1=[0:ts:23.2];
end
for i=1:size(Y64)
   Y64A(i,1:3)=Y64(1,1:3)-Y64(i,1:3);
   Y64B=Y64A.*(1.5e-8);
   Y64C=Y64.*(1.5e-8);
   B64(i)=sqrt((Y64B(i,1)).^2+(Y64B(i,2)).^2+(Y64B(i,3)).^2);
   T64=B64.*mu(3);
   t2=[0:ts:19.65];
end
for i=1:size(Y32)
   Y32A(i,1:3)=Y32(1,1:3)-Y32(i,1:3);
   Y32B=Y32A.*(1.5e-8);
   Y32C=Y32.*(1.5e-8);
   B32(i)=sqrt((Y32B(i,1)).^2+(Y32B(i,2)).^2+(Y32B(i,3)).^2);
   T32=B32.*mu(4);
   t3=[0:ts:14.85];
end
```

```
for i=1:size(Y16)
   Y16A(i,1:3)=Y16(1,1:3)-Y16(i,1:3);
   Y16B=Y16A.*(1.5e-8);
   Y16C=Y16.*(1.5e-8);
   B16(i)=sqrt((Y16B(i,1)).^2+(Y16B(i,2)).^2+(Y16B(i,3)).^2);
   T16=B16.*mu(5);
   t4=[0:ts:20.3];
end

for i=1:size(Y8)
   Y8A(i,1:3)=Y8(1,1:3)-Y8(i,1:3);
   Y8B=Y8A.*(1.5e-8);
   Y8C=Y8.*(1.5e-8);
   B8(i)=sqrt((Y8B(i,1)).^2+(Y8B(i,2)).^2+(Y8B(i,3)).^2);
   T8=B8.*mu(6);
   t5=[0:ts:21.2];
end
for i=1:size(Y4)
   Y4A(i,1:3)=Y4(1,1:3)-Y4(i,1:3);
   Y4B=Y4A.*(1.5e-8);
   Y4C=Y4.*(1.5e-8);
   B4(i)=sqrt((Y4B(i,1)).^2+(Y4B(i,2)).^2+(Y4B(i,3)).^2);
   T4=B4.*mu(7);
   t6=[0:ts:18.45];
end
for i=1:size(Y2)
   Y2A(i,1:3)=Y2(1,1:3)-Y2(i,1:3);
   Y2B=Y2A.*(1.5e-8);
   Y2C=Y2.*(1.5e-8);
   B2(i)=sqrt((Y2B(i,1)).^2+(Y2B(i,2)).^2+(Y2B(i,3)).^2);
   T2=B2.*mu(8);
   t7=[0:ts:19.25];
end
for i=1:size(Y1)
   Y1A(i,1:3)=Y1(1,1:3)-Y1(i,1:3);
   Y1B=Y1A.*(1.5e-8);
   Y1C=Y1.*(1.5e-8);
   B1(i)=sqrt((Y1B(i,1)).^2+(Y1B(i,2)).^2+(Y1B(i,3)).^2);
   T1=B1.*mu(9);
   t8=[0:ts:22.4];
end
% Capture Data
figure(1)
subplot(2,2,1)
```

```
plot(t,B255),axis([8 14 0 4.5e-4]),grid on
title('Y T-Rod Mag Field (255)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t,T255),axis([8 13 0 0.015]),grid on
title('Y T-Rod Torque (255)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t1,B128),axis([9 16 0 5.5e-4]),grid on
title('Y T-Rod Mag Field (128)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t1,T128),Axis([10 16 0 7.5e-3]),grid on
title('Y T-Rod Torque (128)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(2)
subplot(2,2,1)
plot(t2,B64),axis([9 15 0 5e-4]) ,grid on
title('Y T-Rod Mag Field (64)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t2,T64),axis([9 15 0 4e-3]),grid on
title('Y T-Rod Torque (64)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t3,B32),axis([5 11 0 5.5e-4]) ,grid on
title('Y T-Rod Mag Field (32)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t3,T32),grid on,axis([5 11 0 2e-3])
title('Y T-Rod Torque (32)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(3)
subplot(2,2,1)
plot(t4,B16),axis([9 15 0 4e-4]),grid on
title('Y T-Rod Mag Field (16)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t4,T16),grid on,axis([9 15 0 8e-4])
title('Y T-Rod Torque (16)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t5,B8),axis([9 15 0 2e-4]),grid on
```

```
title('Y T-Rod Mag Field (8)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t5,T8),grid on,axis([9 15 0 2.5e-6])
title('Y T-Rod Torque (8)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(4)
subplot(2,2,1)
plot(t6,B4), axis([9 15 0 1e-4]),grid on
title('Y T-Rod Mag Field (4)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t6,T4),grid on,axis([9 14 0 4.5e-5])
title('Y T-Rod Torque (4)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t7,B2),axis([8 14 0 .5e-4]),grid on
title('Y T-Rod Mag Field (2)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t7,T2),grid on,axis([8 14 0 1.2e-5])
title('Y T-Rod Torque (2)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(5)
subplot(2,2,1)
plot(t8,B1), axis([9 16 0 2.3e-5]),grid on
title('Y T-Rod Mag Field (1)')
xlabel('Seconds'),ylabel('Tesla')
subplot (2,2,2)
plot(t8,T1),grid on,axis([9 16 0 3e-6])
title('Y T-Rod Torque (1)')
xlabel('Seconds'),ylabel('Newton*Meters')

subplot(2,1,2)
semilogy(t,T255,t1,T128,t2,T64,t3,T32,t4,T16,t5,T8,t6,T4,t7,T2,t8,T1)
grid on, axis([5 16 10e-10 5e-2])
title('FM-2 TORQUE RANGE')
xlabel('Seconds'), ylabel('Newton Meters')
legend('255','128','64','32','16','8','4','2','1')
```

```
% LCDR Eric W. Herbert, USN, Naval Post Graduate School, Monterey, CA
% 18 August 2004, Z Torque Rod Magnetic Field and Torque Calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, clc
fs=20;                % sampling frequency
ts=1/fs;              % sampling period
r=13.5/1000;          % rod radius (meters)
A=pi*r^2;             % Area enclosed by each turn in coil
mu_M=33;              % rod dipole moment(amp meters^2) at 145 mA
max_i=145e-3;         % max current through solenoid (amps)
N=mu_M/(max_i*pi*r^2);  % # coils on rod
% current at bit levels 255,128,64,32,16,8,4,2,1
I=[140e-3 70e-3 35e-3 17.5e-3 8.75e-3 4.375e-5 2.1875e-3 1.09375e-3 .546875e-3];
Bits=[255:-1:0];
I_step=linspace(140e-3,.546875e-3,256);
mu=N*A.*I;        % calculated dipole moments as I decreases

% load measured data
Z255=load ('ZMFLD255.txt'); Z128=load('ZMFLD128.txt');Z64=load('ZMFLD64.txt');
Z32=load('ZMFLD32.txt'); Z16=load('ZMFLD16.txt'); Z8=load('ZMFLD8.txt');
Z4=load('ZMFLD4.txt'); Z2=load('ZMFLD2.txt'); Z1=load('ZMFLD1.txt');

% extract magnetic north bearing from measured data
Z255=Z255(:,1:3); Z128=Z128(:,1:3); Z64=Z64(:,1:3);
Z32=Z32(:,1:3); Z16=Z16(:,1:3); Z8=Z8(:,1:3);
Z4=Z4(:,1:3); Z2=Z2(:,1:3); Z1=Z1(:,1:3);

% external magnetic field (Tesla)
B=sqrt((Z255(1,1)).^2+(Z255(1,2)).^2+(Z255(1,3)).^2);
Bfld=B.*(1.5e-8);

% convert induced magnetic field (Tesla) and Torque (Newton meters)
for i=1:size(Z255)
   Z255A(i,:)=Z255(1,:)-Z255(i,:);
   Z255B=Z255A.*(1.5e-8);
   Z255C=Z255.*(1.5e-8);
   B255(i)=sqrt((Z255B(i,1)).^2+(Z255B(i,2)).^2+(Z255B(i,3)).^2);
   T255=B255.*mu(1);
   t=[0:.05:14.1];
end
for i=1:size(Z128)
   Z128A(i,1:3)=Z128(1,1:3)-Z128(i,1:3);
   Z128B=Z128A.*(1.5e-8);
   Z128C=Z128.*(1.5e-8);
   B128(i)=sqrt((Z128B(i,1)).^2+(Z128B(i,2)).^2+(Z128B(i,3)).^2);
```

```
    T128=B128.*mu(2);
    t1=[0:ts:11.65];
end
for i=1:size(Z64)
    Z64A(i,1:3)=Z64(1,1:3)-Z64(i,1:3);
    Z64B=Z64A.*(1.5e-8);
    Z64C=Z64.*(1.5e-8);
    B64(i)=sqrt((Z64B(i,1)).^2+(Z64B(i,2)).^2+(Z64B(i,3)).^2);
    T64=B64.*mu(3);
    t2=[0:ts:13.55];
end
for i=1:size(Z32)
    Z32A(i,1:3)=Z32(1,1:3)-Z32(i,1:3);
    Z32B=Z32A.*(1.5e-8);
    Z32C=Z32.*(1.5e-8);
    B32(i)=sqrt((Z32B(i,1)).^2+(Z32B(i,2)).^2+(Z32B(i,3)).^2);
    T32=B32.*mu(4);
    t3=[0:ts:8.6];
end

for i=1:size(Z16)
    Z16A(i,1:3)=Z16(1,1:3)-Z16(i,1:3);
    Z16B=Z16A.*(1.5e-8);
    Z16C=Z16.*(1.5e-8);
    B16(i)=sqrt((Z16B(i,1)).^2+(Z16B(i,2)).^2+(Z16B(i,3)).^2);
    T16=B16.*mu(5);
    t4=[0:ts:9.05];
end

for i=1:size(Z8)
    Z8A(i,1:3)=Z8(1,1:3)-Z8(i,1:3);
    Z8B=Z8A.*(1.5e-8);
    Z8C=Z8.*(1.5e-8);
    B8(i)=sqrt((Z8B(i,1)).^2+(Z8B(i,2)).^2+(Z8B(i,3)).^2);
    T8=B8.*mu(6);
    t5=[0:ts:8.3];
end
for i=1:size(Z4)
    Z4A(i,1:3)=Z4(1,1:3)-Z4(i,1:3);
    Z4B=Z4A.*(1.5e-8);
    Z4C=Z4.*(1.5e-8);
    B4(i)=sqrt((Z4B(i,1)).^2+(Z4B(i,2)).^2+(Z4B(i,3)).^2);
    T4=B4.*mu(7);
    t6=[0:ts:10.65];
end
```

```
for i=1:size(Z2)
   Z2A(i,1:3)=Z2(1,1:3)-Z2(i,1:3);
   Z2B=Z2A.*(1.5e-8);
   Z2C=Z2.*(1.5e-8);
   B2(i)=sqrt((Z2B(i,1)).^2+(Z2B(i,2)).^2+(Z2B(i,3)).^2);
   T2=B2.*mu(8);
   t7=[0:ts:10.4];
end
for i=1:size(Z1)
   Z1A(i,1:3)=Z1(1,1:3)-Z1(i,1:3);
   Z1B=Z1A.*(1.5e-8);
   Z1C=Z1.*(1.5e-8);
   B1(i)=sqrt((Z1B(i,1)).^2+(Z1B(i,2)).^2+(Z1B(i,3)).^2);
   T1=B1.*mu(9);
   t8=[0:ts:8.5];
end

capture data
figure(1)
subplot(2,2,1)
plot(t,B255),axis([3 9 0 6e-4]),grid on
title('Z T-Rod Mag Field (255)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t,T255),axis([3 9 0 0.02]),grid on
title('Z T-Rod Torque (255)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t1,B128),axis([2 9 0 5.5e-4]),grid on
title('Z T-Rod Mag Field (128)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t1,T128),axis([2 9 0 0.01]),grid on
title('Z T-Rod Torque (128)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(2)
subplot(2,2,1)
plot(t2,B64),axis([5 12 0 6e-4]) ,grid on
title('Z T-Rod Mag Field (64)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t2,T64),axis([5 12 0 5e-3]),grid on
title('Z T-Rod Torque (64)')
xlabel('Seconds'),ylabel('Newton*Meters')
```

```
subplot(2,2,3)
plot(t3,B32),axis([2 8 0 5.5e-4]) ,grid on
title('Z T-Rod Mag Field (32)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t3,T32),grid on,axis([2 8 0 2.5e-3])
title('Z T-Rod Torque (32)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(3)
subplot(2,2,1)
plot(t4,B16),axis([2 7 0 5e-4]),grid on
title('Z T-Rod Mag Field (16)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t4,T16),grid on,axis([2 7 0 1e-3])
title('Z T-Rod Torque (16)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t5,B8),axis([2 7 0 2.5e-4]),grid on
title('Z T-Rod Mag Field (8)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t5,T8),grid on,axis([2 7 0 2.5e-6])
title('Z T-Rod Torque (8)')
xlabel('Seconds'),ylabel('Newton*Meters')

figure(4)
subplot(2,2,1)
plot(t6,B4),axis([2 8 0 1.2e-4]),grid on
title('Z T-Rod Mag Field (4)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,2)
plot(t6,T4),grid on,axis([2 8 0 5.5e-5])
title('Z T-Rod Torque (4)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,2,3)
plot(t7,B2),axis([2 8 0 5e-5]),grid on
title('Z T-Rod Mag Field (2)')
xlabel('Seconds'),ylabel('Tesla')
subplot(2,2,4)
plot(t7,T2),grid on,axis([2 7 0 1.4e-5])
title('Z T-Rod Torque (2)')
xlabel('Seconds'),ylabel('Newton*Meters')
```

```
figure(5)
subplot(2,2,1)
plot(t8,B1),axis([2 7 0 2.3e-5]),grid on
title('Z T-Rod Mag Field (1)')
xlabel('Seconds'),ylabel('Tesla')
subplot (2,2,2)
plot(t8,T1),grid on,axis([2 7 0 3e-6])
title('Z T-Rod Torque (1)')
xlabel('Seconds'),ylabel('Newton*Meters')
subplot(2,1,2)
semilogy(t,T255,t1,T128,t2,T64,t3,T32,t4,T16,t5,T8,t6,T4,t7,T2,t8,T1)
grid on, axis([0 12 10e-10 5e-2])
title('FM-3 TORQUE RANGE')
xlabel('Seconds'), ylabel('Newton Meters')
legend('255','128','64','32','16','8','4','2','1')

figure(6)
semilogy(t7,B2), grid on, axis([2 11 10e-9 10e-5])
title('FM-3 Residual Magnetism Example')
xlabel('Seconds'), ylabel('Newton Meters')
```

REFERENCES

[1] James R. Wertz, *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Boston, 1990.

[2] Kimmo Korhonen, "Geomagnetic Models and the IGRF Applet," Master's Thesis, Finnish Meteorlogical Institute, Finland, 2000, http://www.geo.fmi.fi/MAGN/igrf/. Last accessed Sept. 2004.

[3] Marcel J. Sidi, *Spacecraft Dynamics and Control*,Cambridge University Press, New York, 1997.

[4] Barry S. Leonard, Notes for AA3818 (Spacecraft Dynamics and Control), Naval Post-graduate School, Monterey, CA, 2003. (Unpublished)

[5] James R. Wertz, and Wiley J. Larson, *Space Mission Analysis and Design*, 3d ed., Kluwer Academic Publishers, Boston, 1999.

[6] Barry S. Leonard, "NPSAT1 Magnetic Attitude Control System," *Proceedings of the 16th Annual AIAA/USU Conference on Small Satellites*, Paper SSC02-V-7, pp. 203–230, Logan, Utah, Aug. 2002.

[7] Francois Martel, Pal K. Parimal, and Mark Psiaki, "Active Magnetic Control Systems for Gravity Gradient Stabilized Spacecraft," *Second Annual AIAA/USU Conference on Small Satellites.* pp. 103–124, Logan UT, Sept. 1998.

[8] "NPSAT1 Magnetic Torquer Acceptance Test Procedure," Doc. No. MC-02-1898, Issue 4 R2, Microcosm, El Segundo, CA, 20 March, 2002.

[9] Barry S. Leonard, "NPSAT ADCS Interim Review," Naval Postgraduate School, Monterey, CA, August 2001. (Unpublished)

[10] National Oceanic and Atmospheric Administration, "International Geomagnetic Reference Field – Epoch 2000, Revision of the IGRF for 2000-2005," National Geophysical Data Center, www.http:// ngdc.noaa.gov/IAGA/vmod/igrf.html. Last accessed June 2004.

[11] Alexander Schmidt, "NPSAT1 Attitude Control Subsystem Hardware-in-the-Loop Simulation," Master's Thesis, Naval Postgraduate School, Monterey, CA, 2003.

[12] HONEYWELL, "HMR2300: Smart Digital Magnetometer," http://www.ssec.honeywell.com/magnetic/magnetometers.html Last accessed Sept. 2004.

[13] Mathworks, *Real Time Workshop: For Use With Simulink,* Mathworks, Inc, Version 5, Natick, MA, 2002.

[14] David Halliday, Robert Resnick, and Jearl Walker, *Fundamentals of Physics*, 6th ed., John Wiley and Sons, Inc., New York, 2001.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center
Ft. Belvoir, Virginia

Dudley Knox Library
Naval Postgraduate School
Monterey, California